
ESTUDI DE CASOS PRÀCTICS D'ENGINYERIA DE FLUIDS MITJANÇANT PROGRAMARI LLIURE

MEMÒRIA

Autor:

Jacinto Vidal Noguera

Director:

Pedro Javier Gamez Montero



UNIVERSITAT POLITÈCNICA DE CATALUNYA
GRAU EN ENGINYERIA MECÀNICA

9 DE JUNY 2019

Índex

1	Objectiu	7
2	Abast	8
3	Requeriments	9
4	Justificació i utilitat	10
5	Antecedents	11
6	Estat de l'art	12
7	Plantejament de varies opcions sobre el software a utilitzar	14
8	Instal·lació del programari	16
8.1	Descàrrega del Linux	16
8.2	Descàrrega del OpenFOAM versió 6	19
8.3	Descàrrega de diferents programes complementaris	20
8.3.1	Xming	21
8.3.2	Descàrrega del Programa ParaView	22
8.3.3	Descàrrega del programa gedit	23
8.3.4	Descàrrega del programa Salome	23
8.3.5	Esquema general del procediment a seguir per un cas.	24
9	Capítol 1: Estudi d'un flux laminar entre dos plans paral·lels	25
9.1	Proposta del problema	25
9.2	Cas 1: Plaques paral·leles amb una d'elles amb velocitat constant i flux laminar	25
9.2.1	Desenvolupament físic	26
9.2.2	Resolució amb OpenFoam i Pre-processing	27
9.2.3	Creació de la malla	35
9.2.4	Executar la solució	37
9.2.5	Postprocessing i executació de la aplicació	37
9.3	Cas 2: plans paral·lels amb un gradient de pressió (plane-Poiseuille flow) . .	41
9.3.1	Resolució analítica	42
9.3.2	Preprocessing i execució del programa	43
9.3.3	Postprocessing	44
9.4	Cas 3: Plans paral·lels amb moviment ambdós amb la mateixa direcció però en sentit contrari	52
9.4.1	Procediment físic	52
9.4.2	Pre-processing	53
9.4.3	Postprocessing	55
10	Capítol 2: Estudi d'un flux per una canonada	58
10.1	Cas 1: Flux d'una canonada amb un gradient de pressions.	58
10.1.1	Resolució analítica	59
10.1.2	Resolució amb OpenFoam	60

10.1.3	Postproces	64
10.2	Cas 2: Flux en una canonada anular	71
10.3	Resolució analítica	71
10.3.1	Preprocés amb OpenFOAM	73
10.4	Cas 3 : Secció anular amb el cilindre interior amb velocitat angular	78
10.4.1	Resolució analítica	79
10.4.2	Preprocés amb OpenFoam	80
11	Capítol 3: Estudi d'un flux per un colze	89
11.1	Cas1: Estudi d'un colze simple	89
11.1.1	Proposta del problema	89
11.1.2	Generació de la geometria del colze 3D amb el programa Salome.	90
11.1.3	Creació de la malla amb snappyHexMesh	94
11.1.4	Resolució amb un model turbulent	98
11.1.5	Resolució amb un model laminar	104
11.1.6	Comparació dels del cas turbulent amb el cas laminar amb paraview	105
11.2	Cas 2: Estudi d'un colze amb dues entrades	106
11.2.1	Proposta del problema	106
11.2.2	Generació de la geometria	107
11.2.3	Creació del cas i de la malla	109
11.2.4	Execució i solucions del cas	111
12	Capítol 4: Estudi d'un flux d'aire per un F1	114
12.1	Proposta del problema	114
12.2	Adaptació de la geometria.	115
12.3	Preprocés	116
12.3.1	Creació de la malla background	116
12.3.2	Creació del fitxer snappyHexMeshDict	117
12.3.3	Generació dels altres arxius	118
12.4	Visualització de la malla i execució del cas	119
12.5	Postprocés	120
12.5.1	Streamlines	121
12.5.2	Càlcul de la força d'arrossegament i força de sustentació amb Paraview	123
13	Síntesi de resultats	127
14	Implicació Mediambiental	129
15	Planificació	130
16	Pressupost	133
17	Futurs projectes	135
18	Conclusions	136
	Bibliografia	138

Índex de figures

6.1	Anàlisi del camp de velocitats, amb streamlines.	13
7.1	Inversió de millions de dolars en funció del temps [22]	14
7.2	Softwares més utilitzats [23]	14
8.1	Estructura que segueix el programa [4]	16
8.2	Selecció del mode programador	17
8.3	Selecció del subsistema de Windows per a Linux	18
8.4	Descàrrega de Linux	18
8.5	Procediment per descarregar el programa de postprocès	22
8.6	Pantalla inicial del programa ParaView	23
8.7	Procediment general a seguir als següents casos	24
9.1	Esquema del cas 1	26
9.2	Bloc simple [4]	29
9.3	Cell expansion ratios [4]	29
9.4	Carpetes i fitxers que ha de tenir el cas	35
9.5	Malla del cas	37
9.6	Camp de velocitats	38
9.7	Camp de pressions	38
9.8	Generació del perfil de velocitat en funció de l'eix y	39
9.9	Generació del perfil de velocitats representant-ho amb vectors damunt la superfície.	40
9.10	Al costat de la imatge hi ha els valors aconsellables per tenir una bona visualització.	41
9.11	Representació gràfica del segon cas	41
9.12	Representació del camp de velocitats del cas 2.	44
9.13	Representació del camp de pressions del cas 2.	45
9.14	Representació de la velocitat en funció de la distància de l'eix Y.	45
9.15	Representació de la pressió en funció de la distància de l'eix x.	45
9.16	Finalització del procediment	47
9.17	Visualització amb forma de camp de la pressió total	48
9.18	Visualització amb forma de vectors de la pressió total, $\frac{P}{\rho} + \frac{ U ^2}{2}$	48
9.19	Esquema de fitxers del model Turbulent	50
9.20	Esforços tangencials	51
9.21	Variació dels esforços tangencials a l'eix de les y	51
9.22	Dues plaques paral·leles mòbils amb la mateixa direcció i mòdul però diferent sentit.	52
9.23	Esquema de la nova malla	53
9.24	Nova malla	54
9.25	Visualització del perfil de velocitats amb valor absolut	55
9.26	Visualització de la variable U_X	56
9.27	Finalització del procediment	56
9.28	Visualització de StreamLines	57
10.1	Esquema del cas	58

10.2	Esquema de fitxers i carpetes	60
10.3	Resperesntació de la forma de la malla indicant les seves regions	61
10.4	Esquema de funcionament de la funció inletOutlet [9]	63
10.5	Malla generada	64
10.6	Camp de velocitats del cas	65
10.7	Camp escalar de pressió	65
10.8	Variació de la pressió al llarg de la canonada	66
10.9	Variació de la pressió a l'inlet	66
10.10	Variació de la pressió en l'eix de la y a la regió inlet.	67
10.11	Imatge de linux amb el valor del cabal	68
10.12	Selecció dels valors de les cel·les	68
10.13	Gràfica del residus en funció del temps.	70
10.14	Esquema del cas 2.2	71
10.15	Resultats de velocitat	76
10.16	Gràfica de la variació de la velocitat en funció de l'eix de les y	76
10.17	Visualització dels vectors	76
10.18	Gràfica de la variació de la pressió en funció del l'eix z	77
10.19	Resultat del cabal en m^3/s	78
10.20	Esquema del problema resolt, les unitats longitudinals en mm.	79
10.21	Esquema de fitxers del cas 3. En verd apareixen les carpetes mentre que en blau són els fitxers	81
10.22	Esquema de la malla a crear, on apareixen el nom dels punts i de les regions.	81
10.23	Solució del perfil de velocitats	84
10.24	Variació de la velocitat en funció del y	85
10.25	86
10.26	Comparació de la solució analítica i la solució del openfoam amb el solver simpleFoam	86
10.27	Visualització dels vectors	87
10.28	Gràfica dels esforços tangencials en l'eix z.	87
11.1	Dades per resoldre el problema. Les dades geomètriques en mm	89
11.6	Generació del disc	92
11.7	Extrussió d'un disc per un camí	92
11.8	Funció explode	93
11.11	El sòlid creat i la malla del blockMesh	96
11.12	Malla executada per snappyHexMesh	97
11.13	Malla amb l'arxiu "surfaceFeatureExtractDict"	98
11.14	Esquema model turbulent amb simpleFoam	100
11.15	Gràfica dels residuals de les diferents magnituds	102
11.16	Gràfica dels residuals aplicant un factor de relaxació	102
11.17	Resultats de la velocitat del colze simple amb un tall amb l'eina "clip"	103
11.18	Pressió a la regió outlet	104
11.19	Pressió a la regió inlet	104
11.20	A l'esquerra apareix el cas turbulent i a la dreta apareix el cas laminar amb una mateixa escala	106
11.21	Colze amb dues entrades, les dades geomètriques estan en mm.	106
11.22	Generació d'un punt damunt una superfície.	107
11.23	Generació del disc	108
11.24	Extrucció	108
11.25	Utilització de la funció fuse	109

11.26	Nom de les cares i l'agrupació de cares per formar walls.	109
11.27	Malla del colze amb dues entrades	110
11.28	Camp de velocitats del cas en m/s	111
11.29	Resultat de la pressió en m^2/s^2	111
11.30	Cabal d'outlet en m^3/s	112
11.31	Cabal inlet1 en m^3/s	112
11.32	Cabal d'inlet2 en m^3/s	112
12.1	Ús de l'eina de mesura amb paraview	114
12.2	Icones per tal d'adaptar la geometria a l'usuari	115
12.3	Geometria final	116
12.4	Utilitzant la funció sphere en el programa paraview	117
12.5	temps=1, generat per castellated mesh	119
12.6	temps=2, generada per snnapyHexMesh	119
12.7	Malla completa	120
12.8	Camp de pressions damunt el sòlid F1	121
12.9	Camp de velocitats damunt el sòlid F1	121
12.10	Visualització de la malla i la velocitat.	122
12.11	Procediment per crear línies de corrent amb Streamtracer	122
12.12	Visualització final de les línies de corrent	123
12.13	Ús de la funció extractBlock	123
12.14	Funció ExtractSurface	124
12.15	Generació del vector normal de les cel·les i visualització dels vectors en Y. .	124
12.16	Visualització de la força de sustentació,	125
12.17	Força de sustentació en Newtons.	125
12.18	Força d'arrossegament en Newtons.	126
15.1	Llistat de tasques	130
15.2	Diagrama de Gantt	131
15.3	Gràfica de la densitat de treball	132
15.4	Dedicació d'hores per grups	132

Índex de taules

9.1 Unitats bàsiques [4]	31
10.1 Tipus de formes dintre del diccionari BlockMeshDict [4]	62
14.1 Càlcul dels kg CO_2	129
16.1 Referent a la quantificació econòmica a la gestió de la informació	133
16.2 Referent a la quantificació econòmica dels pre-processos	133
16.3 Referent als postprocessos	134
16.4 Total del pressupost	134

1. Objectiu

L'objectiu principal d'aquest estudi és la creació de casos pràctics de l'enginyeria de fluids i de la resolució aquests casos, mitjançant simulacions utilitzant programari lliure. Des d'una altra perspectiva pràctica, serà una guia d'iniciació en cos de projecte per als estudiants d'enginyeria que vulguin iniciar-se dintre l'àmbit de la dinàmica de fluids computacional també conegut com a CFD, ja que aquest temari no es toca dintre del grau.

Un cop el tutorial principal de la guia oficial s'ha finalitzat entra en joc la base d'aquest estudi. Compost per una guia d'instal·lació i quatre capítols on apareixeran diferents casos bàsics resolts. [5]

2. Abast

L'abast d'aquest estudi inclourà:

- Diferents instruccions per l'ús de linux a nivell d'usuari.
- Resolució de diferents problemes relacionats amb la mecànica de fluids mitjançant programari lliure, més concretament amb l'ús d'Openfoam. També Salome com a plataforma per la generació de geometries i Paraview per visualitzar els resultats.
- Ajuda amb la creació de l'estructura d'arxius que ha de tenir cada cas per una correcta interpretació del programa.
- Diferents mètodes a l'hora de creació de les malles.
- Comentaris de les funcions bàsiques per la creació d'una malla.
- Exemples de codi per una millor interpretació.
- Diferents formes de manipular la malla i la seva comprovació.
- Explicacions del control de paràmetres de lectura i escriptura del programa, per una millor resolució i comprensió dels diferents casos.
- Introducció al programa de les propietats físiques del fluid.
- Descripció sobre la generació de diferents condicions de contorn i condicions inicials per cada cas.
- L'ús de diferents "solvers".
- Explicacions de com visualitzar els resultats calculats per al programa.
- Utilització de diferents funcions de postprocès que s'inclouen al programa.
- Interpretació dels resultats amb el programa Paraview.

3. Requeriments

Per tal de poder executar aquest estudi és necessari la utilització del programari adequat. Tot el programari utilitzat serà lliure. Per tant els requeriments d'aquest projecte per tal de poder complir amb l'abast de l'estudi són:

- La descàrrega del programari, necessari com és en aquest cas de **Openfoam** [4], per solucionar el problema. **ParaView** [15] per visualitzar els resultats generats per Openfoam. Plataforma per generar geometries i malles, anomenada **Salome** [17].
- Coneixements bàsics de mecànica de fluids, per tal de poder entendre les funcions que s'apliquen al programa i també per fer una interpretació correcta dels resultats.
- Coneixements bàsics de programació, ja que s'implementarà dins dels arxius un codi que ha de llegir el programa.

4. Justificació i utilitat

Els grans avanços en la tecnologia han induït també grans millores en la resolució de problemes d'enginyeria. Un exemple d'aquestes millores es troba en el mètode dels volums finits on juga un paper fonamental el CFD. La complexitat d'aquests mètodes fan que la seva divulgació en àmbits universitaris no sigui molt extensa. Al mateix temps, en ser conceptes relativament innovadors sovint no s'acaben d'entendre. Aquests dos factors són el principal motiu de la dificultat de la matèria.

També apareix un problema relacionat amb els programes lliures, ja que aquests no tenen tantes facilitats com un programari comercial. Alguns exemples d'aquestes dificultats vénen determinats pels tutorials perquè deixen a la intuïció conceptes o passos a seguir. A més no ajuda el fet que Openfoam no disposa d'una interfície gràfica com altres programes a més treballa a partir de carpetes i arxius. En conseqüència s'ha d'acabar acudint a internet on la informació és massa específica i mai s'acaba d'adaptar al que seria l'aprenentatge. Per altra banda totes les guies disponibles es troben centrades en eines molt avançades.

Per això es creu que amb l'execució d'aquest estudi que té com a base la creació i explicació de casos pràctics a l'enginyeria de fluids, mitjançant una guia d'iniciació permetrà que un nou usuari, estudiant d'enginyeria de fluids, pugui gaudir d'una major facilitat a l'hora d'iniciar-se dintre l'àmbit de la mecànica de fluids computacional i profunditzar dintre del programari Openfoam®.

5. Antecedents

Aquest estudi començarà del nivell més bàsic. Per tal de donar nocions sobre el coneixement de la mecànica computacional de fluids. Això sí, cal tenir una base sobre mecànica de fluids per poder interpretar resultats o fins i tot per poder comprovar-los.

Cal remarcar, que es parteix dels fonaments del CFD, ja que en el grau d'enginyeria mecànica, no s'ha fet menció sobre aquests coneixements. Per altra banda de forma autodidàctica la branca del CFD pot arribar a ser una mica exòtica. Per això cal tenir una base consolidada de mecànica de fluids, per poder progressar adequadament dintre l'àmbit del CFD. A més els coneixements adquirits al grau de mecànica de fluids són una mica distints dels utilitzats en aquesta guia. Augmentant així la dificultat d'entendre el funcionament del programa.

Per tal de poder iniciar-se dintre del CFD i poder assolir els coneixements d'una forma més autònoma s'ha utilitzat una guia anomenada “foam-house”. [1]

6. Estat de l'art

El CFD ha experimentat un gran avanç a l'hora de resoldre problemes englobats dins l'àmbit de la mecànica de fluids. El CFD o també anomenat dinàmica de fluids computacional es va inicialitzar als anys 60, encara que els primers èxits varen començar als anys 70. El CFD és una combinació de física, mètodes numèrics i informàtica. A causa d'aquesta revolució, varen començar a sorgir les primeres companyies que es dedicaven al disseny de software pel càlcul dels volums finits una d'elles i també de les més conegudes és **Ansys Inc** creada a Pennsilvània i fundada per John A. Swanson. [13]

Fins als mitjans anys 80 no es va començar a centrar amb les equacions de Navier-Stokes per fluids viscosos. A causa de tot aquest desenvolupament, es varen crear models numèrics de fluxos turbulents com serien Direct Numerical Simulation(DNS) i el Large Eddy Simulation(LES).

A mesura del pas del temps anaven sorgint la demanda de solucions cada vegada més complexes. Aquest fet provoca que no només vagin acompanyades d'un desenvolupament de nous mètodes numèrics sinó també de noves tècniques de la generació de la malla. Per als fets anteriors, es produeix un gran creixement de la utilització d'aquests mètodes i durant la dècada dels 90, es varen començar a aplicar a la indústria.

A mesura del pas del temps aquest àmbit ha tingut un gran creixement i un dels moments claus ha estat amb l'aparició del programa lliure **OpenFOAM** l'any 2004 creat per Henry Weller, Chris Greenshields i Mattijs Janssens a través de la companyia OpenCFD. Més tard distribuït per la fundació OpenFoam. [12]

En funció del pas del temps fins a l'actualitat, és molt comú referir-se a les eines i tècniques de simulació per l'avanç de la ciència y la tecnologia. En general l'utilització d'aquestes eines s'ha tornat un factor molt important per el desenvolupament de tecnologies cotidianes.

La dinàmica de fluids computacional, va ser desenvolupada dintre del sector aeronàutic i la indústria aeroespacial. Però va tenir tanta repercussió que s'ha convertit amb una eina essencial dintre de diferents àmbits com l'automoció, la generació d'energia, la química, indústria nuclear, marina i molts més.

En els darrers deu anys s'ha afegit la indústria electrònica, on s'utilitza per optimitzar els sistemes d'energia i la transferència de calor, per optimitzar el refredament d'equips electrònics. També s'hi ha incorporat la indústria biomèdica, dintre d'aquest àmbit és una eina de validació per aplicacions mèdiques. A més també, no ha quedat enrere la indústria de la construcció on s'utilitza en el sector HVAC (la ventilació, calefacció i refrigeració amb aires condicionats), també amb la simulació d'incendis i qualitat d'aire.

La mecànica de fluids computacional s'ha incorporat més tard que les eines de mecànica de sòlids y vibracions consolidades als anys 80. Aquest fet és generat per la complexitat de les equacions que s'han de resoldre. Unes d'elles són les equacions de Navier-Stokes que modelen amb gran precisió tot un conjunt de tipus de fluxos com per exemple, incompressibles monofàsics tant siguin turbulents com laminars, fluxos compressibles a grans velocitats i fins i tot fluxos multifàsics. Aquestes equacions combinades amb mètodes numèrics de volums finits són claus pel desenvolupament del CFD on determinen un paper important. [24]

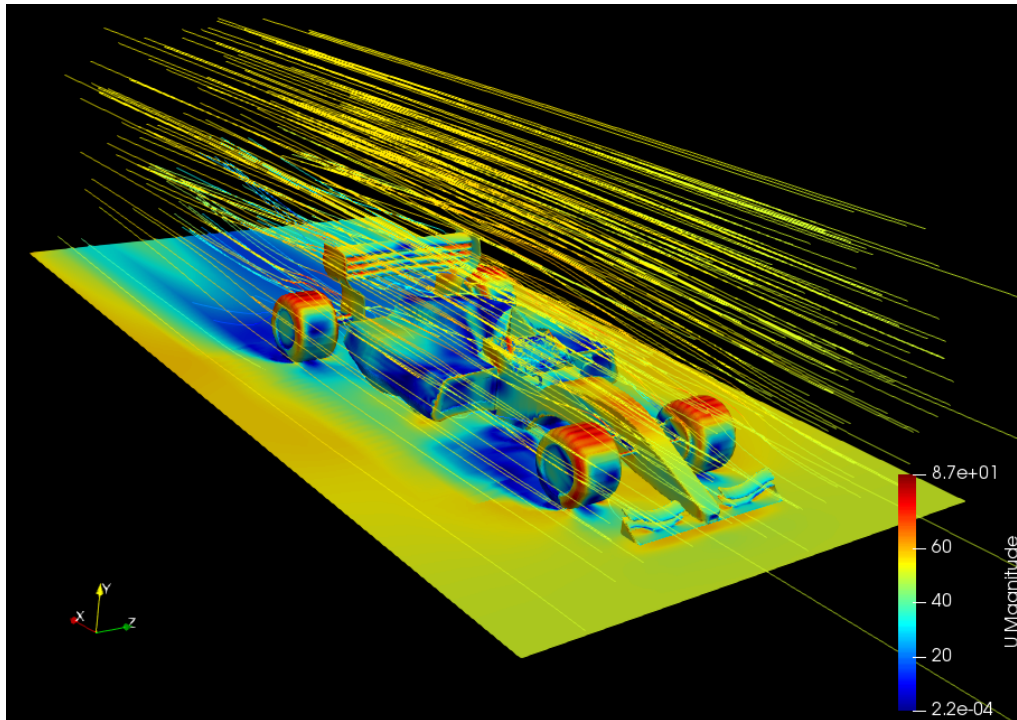


Figura 6.1: Anàlisi del camp de velocitats, amb streamlines.

A causa d'aquesta revolució, actualment conviuen una gran diversitat de softwares per la simulació de CFD. Els més utilitzats per a les simulacions CFD són: el dominant és Ansys Fluent seguit per Star-ccm+ i Flow-3d tots aquests són comercials. S'ha de remarcar que Openfoam, és el programari lliure més utilitzat, a causa de l'elevat cost que tenen les llicències dels softwares CFD comercials. [11]

7. Plantejament de varies opcions sobre el software a utilitzar

El CFD s'està desenvolupant de forma notable i cada vegada agafen més força per tal de demostrar aquest fet s'adjunta una estadística de l'evolució dels milions de dòlars invertits en funció del temps.

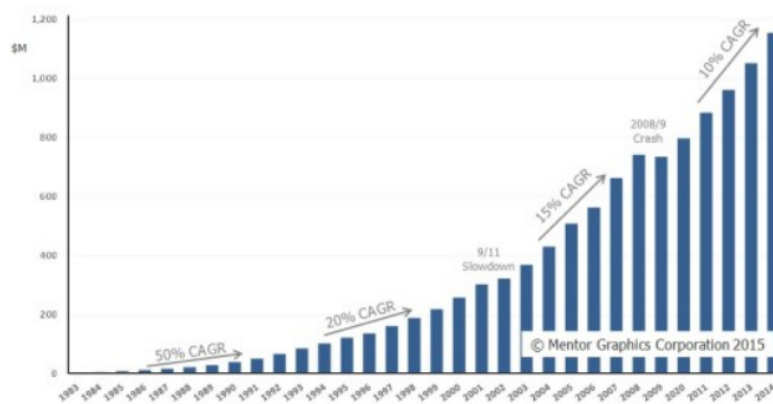


Figura 7.1: Inversió de milions de dolars en funció del temps [22]

El fet que el CFD ha agafat molta força és una realitat. Per tant a l'hora de generar aquest estudi, sorgeix la pregunta: “amb quin software s’ha de realitzar?”. Per tal de respondre a la pregunta i després d’una recerca exhaustiva. S’ha trobat una estadística del 2016 de la pàgina “www.resolvedanalytics.com” que fa una entrevista de quin és el software més utilitzat per als seus usuaris per a la simulació de mecànica de fluids.

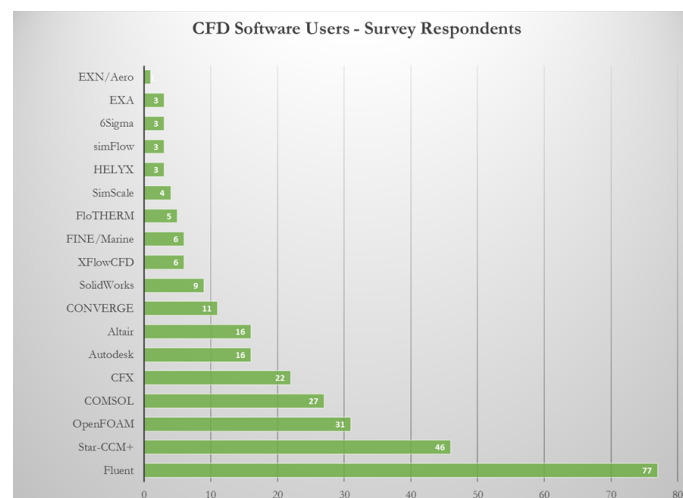


Figura 7.2: Softwares més utilitzats [23]

Com es pot observar a la figura 7.2, els dos més utilitzats Fluent d'Ansys i l'altre Star-CCM+ i el tercer és Openfoam. Els dos primers són comercials. Mentre que Openfoam és un programa que pertany al grup de programes de codi obert o també coneguts com “opensource”. Openfoam és distribuït exclusivament baix la Llicència Publica General (GPL). La llicència GPL dona llibertat de modificar, redistribuir el software i un ús lliure dintre dels termes de llicència. A continuació s'analitzarà quins avantatges té Openfoam respecte dels altres dos:

- La primera i la més important és que els programes de codi obert són gratuïts.
- El fet de que hi hagi una actualització en els programes comercials, pot suposar que el programa quedi inhabilitat i a més comporta una pèrdua de temps i de diners.
- Gran accesibilitat de totes les eines, en canvi en els programes comercials el que fan és vendre els programes en blocs.
- No totes les empreses comercials de software deixen provar el producte.
- Al ser codi obert, les noves funcionalitats solen venir per les necessitats dels usuaris i no per les idees d'un departament de desenvolupament d'una gran empresa.
- La solució d'errors i noves implementacions són corregides en més rapidesa, a causa que el codi obert permet modificar el codi intern i moltes vegades acaben sent solucionats pels mateixos usuaris.
- És uns dels programes més utilitzats dintre de la indústria com s'ha observat 7.2. Openfoam, utilitzat per a moltes institucions. Un exemple clar és que s'utilitza en el departament de fluids de l'ESEIAAT.

8. Instal·lació del programari

Abans de començar la resolució de casos pràctics amb el programari lliure és necessari obtenir els programes que s'utilitzaran. El programa de càlcul utilitzat serà OpenFOAM. Cal destacar que OpenFOAM, utilitza les biblioteques de C++, per crear els seus executables. Cal distingir dues categories: **solvers**, que fan referència a ordres determinades a resoldre un problema dintre de la mecànica de medi continu i **utilities**, que són funcions determinades a la manipulació de dades.

Openfoam com a tal no té un executable i una interfície gràfica com altres programes, sinó que genera els seus fitxers i una vegada generats es poden executar, amb el programa **Paraview**, solucionar i gestionar amb les ordres d'Openfoam, que seran explicats en els següents capítols com generar-los amb la informació corresponent. El que es vol explicar amb la figura 8.1 és l'estructura que segueix el programa Openfoam per l'anàlisi de problemes.

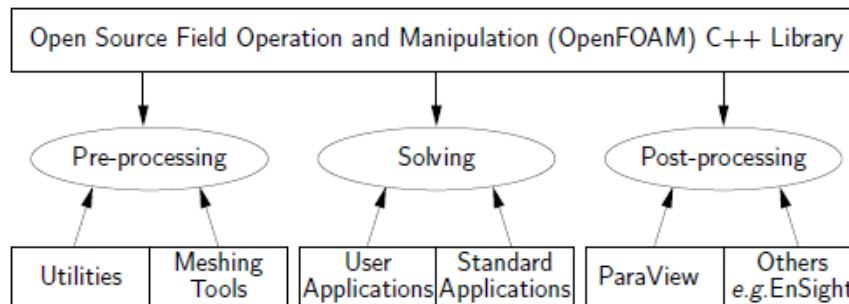


Figura 8.1: Estructura que segueix el programa [4]

El primer és necessari descarregar-se la imatge de linux, ja que és necessari per a l'ús d'Openfoam.

8.1. Descàrrega del Linux

El que es farà en aquest pas és descarregar-se una aplicació de Windows, que deixa utilitzar el Linux dintre del mateix Windows. És útil, ja que l'ús de l'Openfoam només es pot ser utilitzant amb el Linux, s'aporta aquesta solució perquè la gran majoria d'usuaris treballen en Windows. [18]

Primerament cal adaptar l'ordinador per poder descarregar-se l'aplicació, que es faran

seguint els passos següents: [26]

1. Anar a l'opció de **configuració/settings** depenent de l'idioma que té l'ordinador.
2. Dintre de configuració, anar a configuració **per els desenvoladors/for developers**.
3. Apareixeran tres opcions i cal escollir "for developers" o també pot ser que es digui mode programador.

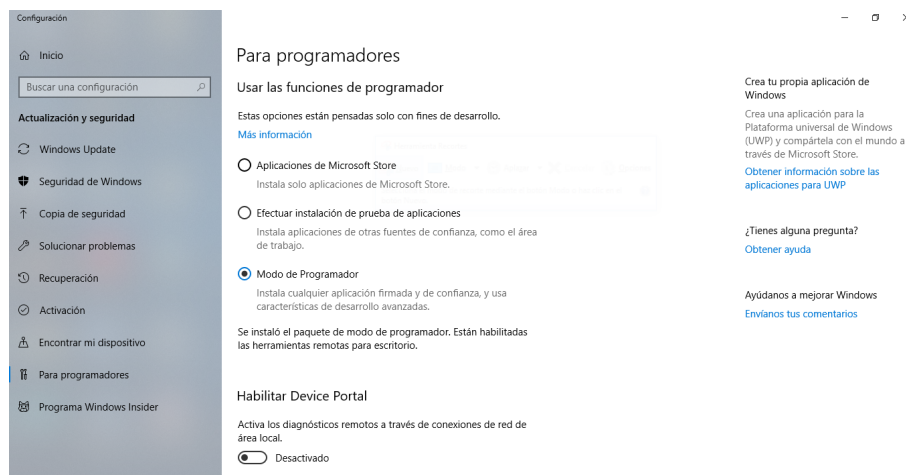


Figura 8.2: Selecció del mode programador

4. Apareixerà una finestra indicant si es desitja canviar a mode programador i es clicarà la paraula "Sí". Cap la possibilitat que tardi una mica ja que necessita carregar.
5. A continuació s'ha d'accedir al panell de control.
6. Clicar a programes.
7. A continuació dintre de **programes i característiques** clicar l'opció Activar o desactivar les característiques de Windows.
8. Apareixerà una finestra que es diu **Windows Featurs** i apareixeran diferents opcions que es poden marcar i s'ha de marcar la que posa **windows subsystem for linux** i clicar "ok". I apareixerà un missatge que cal reiniciar Windows.

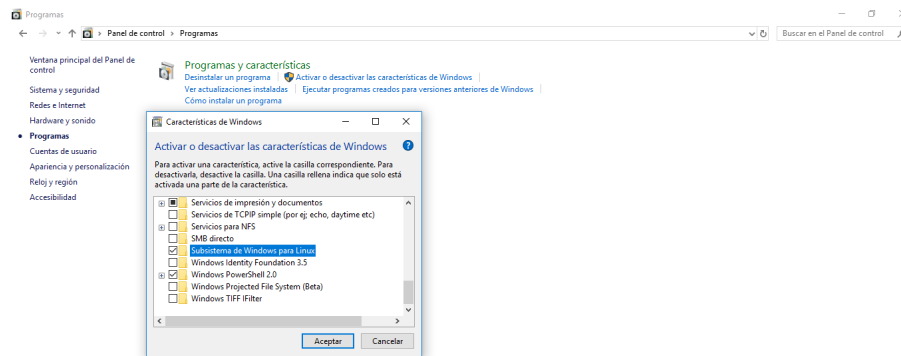


Figura 8.3: Selecció del subsistema de Windows per a Linux

9. Una vegada reiniciat serà convenient anar a la pàgina <https://www.microsoft.com/es-es/windows/windows-10-apps> i descarregar-se l'aplicació **Ubuntu 18.04 LTS** una vegada descarregada es pot executar i demanarà el nom d'usuari i una contrasenya. Amb aquest darrer pas el Linux queda totalment instal·lat.

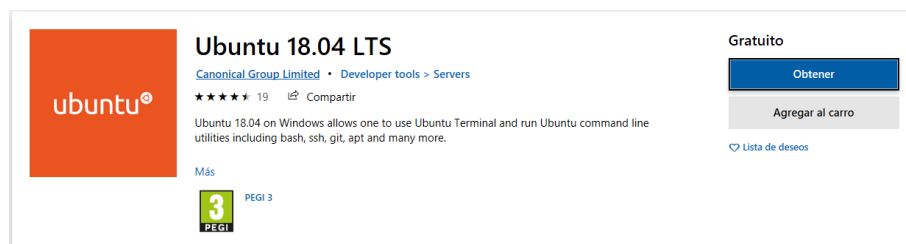


Figura 8.4: Descàrrega de Linux

Per accedir-hi simplement cal buscar al cercador de Windows la paraula **ubuntu** i apareixerà una pantalla emergent, es clicarà damunt i ja s'executarà.

A continuació es crearà una carpeta que vincularà el Windows amb Linux a partir d'aquesta es treballarà sempre des de aquí. Aquesta carpeta tant es podrà generar arxius des de el terminal de Linux com és podran generar i afegir arxius des de Windows. Per tant els passos a seguir són els següents:

1. Crear on és desitja una carpeta. En aquest cas es farà d'exemple creant-la a l'escriptori amb el nom de cfd. Es col·loca a l'escriptori, botó dret, nou, nova carpeta.
2. A continuació s'obri l'aplicació descarregada d'ubuntu. I es cerca el directori on està guarda aquesta carpeta.

```
cd /mnt/c/Users/desktop/cfd
```

És possible que pugui canviar, ja que cada ordinador tot el món el té diferent, el que no canvia és la part de **mnt/c/** l'altre depèn de com l'usuari de l'ordinador tingui guardat. **cd** és la funció que es fa per accedir a dintre d'un directori. I per saber les carpetes que hi ha simplement cal utilitzar la funció **ls**. I trobar el directori.

3. Una vegada s'ha trobat el directori, s'utilitza:

cd

Per tornar a la primera part.

4. Aquí s'utilitza la següent funció (**ln -s**) i a continuació el directori que es vol exportar per exportar la carpeta. Es recorda que s'ha d'estar situat a la carpeta principal, per això anteriorment s'hi ha utilitzat la funció exclusiva **cd**:

ln -s /mnt/c/Users/desktop/cfd

5. Per comprovar-ho simplement cal utilitzar la funció **ls** i es podrà observar que exactament apareix al nom.

Una vegada instal·lat el Linux. Es passarà a la següent part que és la descàrrega de l'Openfoam.

8.2. Descàrrega del OpenFOAM versió 6

Cal recordar que Openfoam és un programa de codi obert i que la seva descàrrega és gratuïta. El procediment està explicat a la pàgina web. En aquest apartat s'explicarà pas per pas. El procediment és el següent: [14] [29]

1. Obrir el terminal Ubuntu.
2. Una vegada obert introduir el següent codi:

```
sudo sh -c "wget -O - http://dl.openfoam.org/gpg.key | apt-  
key add -"  
sudo add-apt-repository http://dl.openfoam.org/ubuntu
```

3. Una vegada descarregat, introduir la següent ordre:

```
sudo apt-get update
```

4. A continuació:

```
sudo apt-get -y install openfoam6
```


5. S'apliquen els patch per OpenFOAM v6, OpenFOAM-6 periòdicament és actualitzat per tant és recomanable descarregar-se els packages per actualitzar-lo de forma automàtica. Amb la següent funció:

```
sudo apt-get update
sudo apt-get upgrade
```

6. El usuari també poden concretament actualitzar openfoam amb l'opció següent:

```
sudo apt-get update
sudo apt-get install --only-upgrade openfoam6
```

7. A continuació es farà una comprovació de si funciona el programa. Introduint al terminal de Linux aquestes ordres:

```
mkdir -p $FOAMRUN
```

8. Copiant el codi anterior es descarrega un exemple de l'Openfoam i es genera una malla i s'executa el "solver". La primera ordre correspon a la carpeta creada anteriorment per als casos

```
cd
cd cfd
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/pitzDaily .
cd pitzDaily
blockMesh
simpleFoam
paraFoam
```

9. Tornar a la pantalla principal amb l'ordre **cp**

10. Indicar al terminal la següent ordre:

```
echo "source /opt/openfoam6/etc/bashrc" >> .bashrc
source ~/.bashrc
```

Primerament es torna a la pantalla inicial, després importarà a la carpeta cfd un tutorial anomenat **pitzDaily**. Si el programa està instal·lat començarà solucionar el problema si no apareix cap mena d'error, és que el programa està ben instal·lat.

8.3. Descàrrega de diferents programes complementaris

En aquest apartat s'instal·laran tots els programes. Amb Xming s'hi inclou bastants de programes. Però durant la resolució sempre es farà directament des de el programa original, ja que és més intuïtiu i a part en el paraview de dintre Xming a vegades no es poden visualitzar únicament les malles, excepte que és molt útil per obrir el programa gnuplot, per això no caldrà la descàrrega del gnuplot.

8.3.1 Xming

En aquest apartat s'explicarà com fer la descàrrega d'aquest programa que ens ajudarà a poder executar tots els programes complementaris com seria paraview, gedit, gnuplot... [16]. Tots aquests es poden executar directament des de la imatge de Linux.

1. Descarregar-se el programa Xming.

<http://www.straightrunning.com/XmingNotes/>

2. Instal·lar el programa.

3. Obrir la imatge de Linux i executar la següent ordre:

```
echo "export DISPLAY=:0" >> .bashrc
source ~/.bashrc
```

4. Cada pic que es vulgui executar el programa paraview s'ha de fer doble clic a la icona Xming i després executar l'ordre: paraFoam.
5. Cada pic que es vulgui crear un arxiu gedit simplement el que s'haurà de fer és al terminal de linux introduir la funció: gedit. Anteriorment s'hi ha d'haver fet clic a Xming.

Es recorda que abans d'executar una ordre que obri un programa extern des de la imatge de Linux s'ha de fer un doble clic damunt de la icona del programa **Xming**. A continuació s'explica també com descarregar el programa Paraview i gedit des de Windows. Així ens permet treballar sense tenir la imatge de Linux oberta, si només es desitja obrir el paraview, per observar un treball ja fet, si anteriorment s'ha creat un arxiu amb extensió foam. En aquesta guia s'intentarà fer tot des del paraview descarregat amb el Windows.

Per obrir paraview des de Linux amb l'ordre (anteriorment fer doble clic a Xming):

```
paraFoam
```

Paraview del Windows crear un fitxer .foam amb la següent ordre:

```
paraFoam -touch -builtin
```

Obrir el programa paraview i cercar l'arxiu creat que estarà ubicat dintre de la carpeta del cas.

8.3.2 Descàrrega del Programa ParaView

Aquest programa és un programa que s'utilitza com a eina del postprocés, per analitzar els resultats que s'han generat en el conjunt de fitxers de després de la simulació. En els pròxims apartats s'explicarà com utilitzar-lo. Cal destacar que no només es pot utilitzar per al postprocés del programa Openfoam sinó que també per altres programes de simulació. [15]

Per instal·lar-lo és molt senzill, únicament cal anar a la pàgina web de paraview i descarregar la versió més actual de Windows.

1. Una vegada accedit a la pàgina web oficial de ParaView, clicar l'apartat de **Download**
2. Actualment la versió més actual és **v5.6**.
3. Seleccionar el sistema operatiu corresponent, en aquest cas és Windows.
4. Clicar damunt el primer arxiu zip.

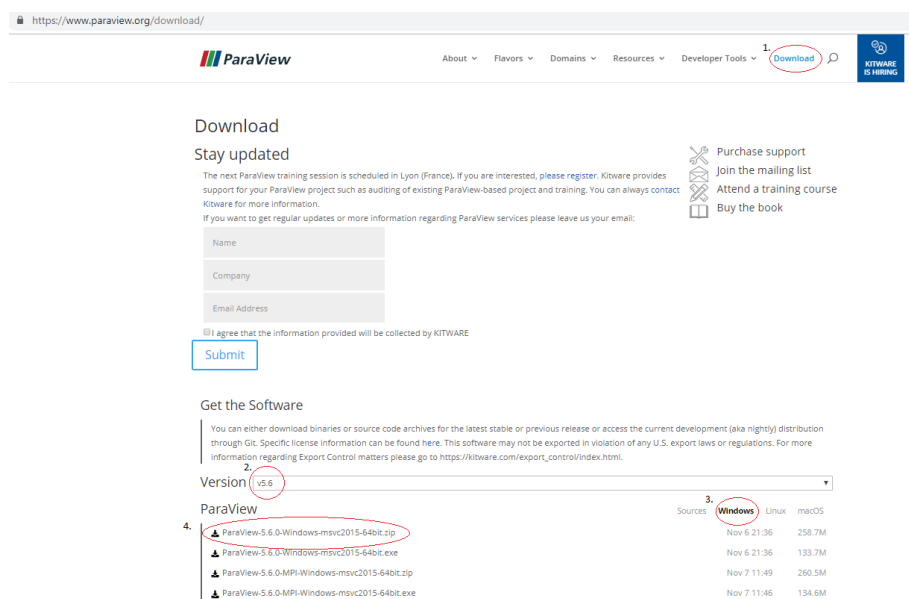


Figura 8.5: Procediment per descarregar el programa de postprocés

Una vegada descarregat simplement cal, descomprimir dins una carpeta aquest arxiu. Aquesta carpeta es pot guardar on es desitgi i amb el nom que l'usuari trobi oportú. Per executar el programa simplement cal anar a la carpeta, descomprimida → bin i dintre d'aquesta carpeta fer doble clic damunt el fitxer que s'anomena **paraview.exe**. També es pot cercar en el cercador de Windows posant **paraview**. Una vegada s'ha executat apareixerà una finestra com aquesta:

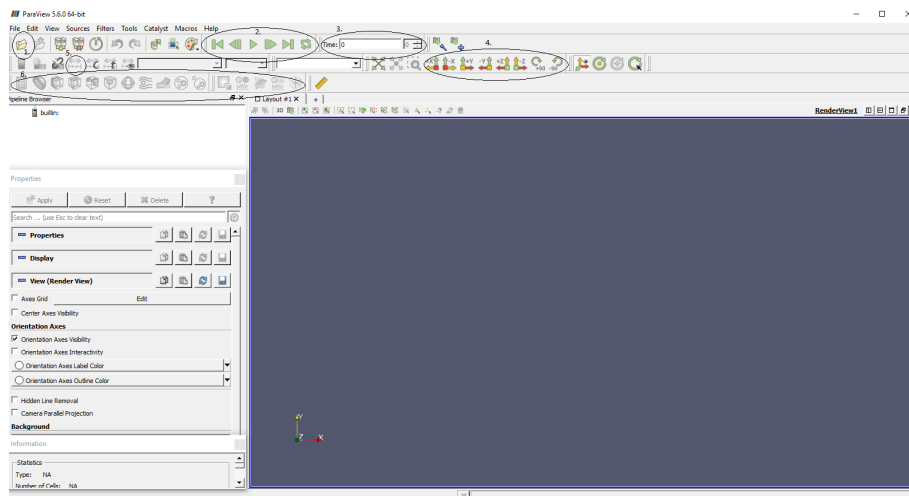


Figura 8.6: Pantalla inicial del programa ParaView

1. Recerca i obertura de fitxers generats.
2. Controladors fitxers generats per OpenFoam.
3. Indicador del nombre de fitxers i indicador del fitxer referent a les carpetes que s'està visualitzant.
4. Diferents vistes per visualitzar la malla o els resultats.
5. A mesura que es desplaça pels diferents fitxers, per observar l'escala correctament cal clicar aquesta icona per refrescar l'escala.
6. Diferents funcions per obtenir la visualització dels resultats.

8.3.3 Descàrrega del programa gedit

Aquest programa és un editor de text lliure. Aquest editor es caracteritza principalment per la seva facilitat d'ús. En aquest estudi, s'utilitzarà per editar o crear els diferents arxius per definir les diferents característiques dels casos estudiats. [19]

Simplement cal anar a la pàgina web oficial i descarregar la versió per Windows.

8.3.4 Descàrrega del programa Salome

Aquest programa s'utilitzarà per crear geometries complexes. Encara que té moltes altres funcions com per exemple, la generació de malles.

Per instal·lar-lo simplement cal anar a la pàgina web oficial [17] i descarregar la versió més recent per a Windows. Una vegada descarregat, obrir la carpeta de Salome i per executar-lo seguir les instruccions del arxiu “Readme”. Per obrir el programa únicament cal executar l'arxiu “run_Salome.bat”. Si demana permisos, fent referència a la seguretat no fa falta obrir-los. A continuació ja s'obrirà.

8.3.5 Esquema general del procediment a seguir per un cas.

Una vegada finalitzat tot aquest procediment s'ha de tenir clar el següent esquema per poder realitzar els casos que vénen a continuació:

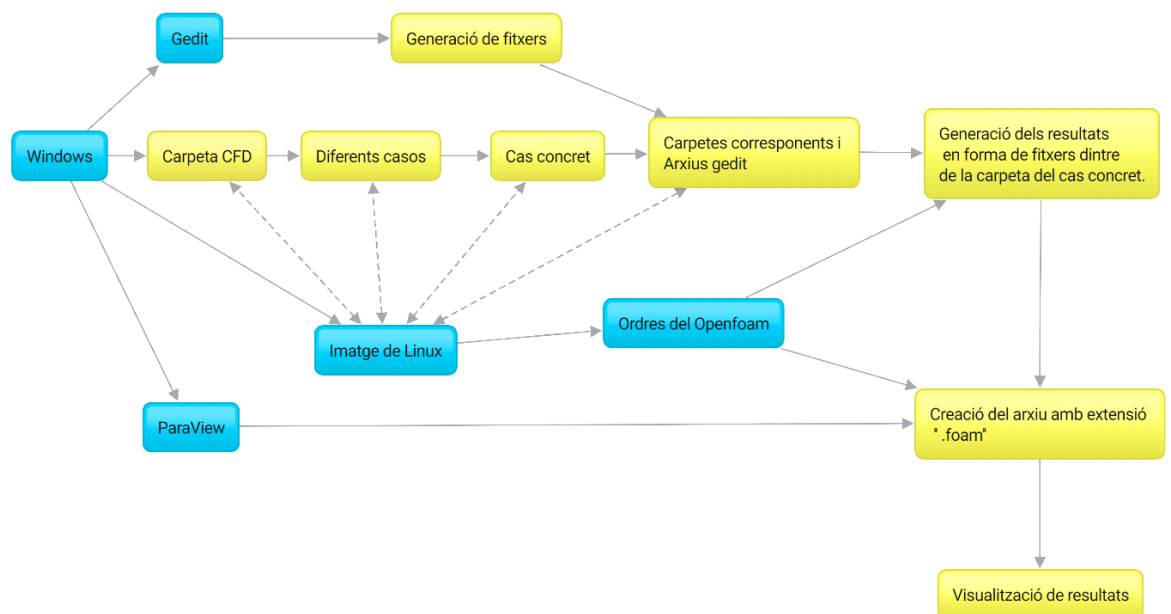


Figura 8.7: Procediment general a seguir als següents casos

Aquest esquema serveix per crear a l'usuari d'un esquema mental d'un cas general, també la ubicació d'on es troba cada programa i el procediment a seguir de forma més general possible. En blau es representa els programes, mentre que en groc són carpetes o arxius. Les línies discontinües representen que hi ha una interacció, mentre que les línies contínues representen accions a seguir.

9. Capítol 1: Estudi d'un flux laminar entre dos plans paral·lels

9.1. Proposta del problema

Aquest tutorial consistirà en l'estudi d'un flux que passa entre dues plaques paral·leles separades per una distància i que una d'elles es mou respecte a l'altre a velocitat constant, aquest cas és conegut com pel flux de Couette. A més es faran diferents modificacions d'aquest cas per tal d'ensenyar una millor interpretació del codi que utilitza el programa. També es faran ús de les funcions bàsiques del programari per tal familiaritzar-se amb el programa Openfoam. A més es farà ús de programari de post-processing, utilitzant el programa ParaView per tal d'aprendre les diferents eines que ens proporciona.

Els dos primers casos s'han extret d'una altra guia però s'adaptaran els casos de la nova versió d'Openfoam [1] [4].

9.2. Cas 1: Plaques paral·leles amb una d'elles amb velocitat constant i flux laminar

Aquest cas consistirà en una placa amb velocitat constant i una altra fixa i entre elles un fluid no hi haurà cap mena de gradient de pressions $\frac{\partial p}{\partial x} = 0$.

Asumirem que:

- Flux incompressible.
- Flux laminar.
- Fluid newtonià.
- $\frac{\partial w}{\partial z} = 0$ (dues dimensions).
- Es menysprearan efectes produïts per la gravetat.
- Flux estacionari $\frac{\partial}{\partial t} = 0$.

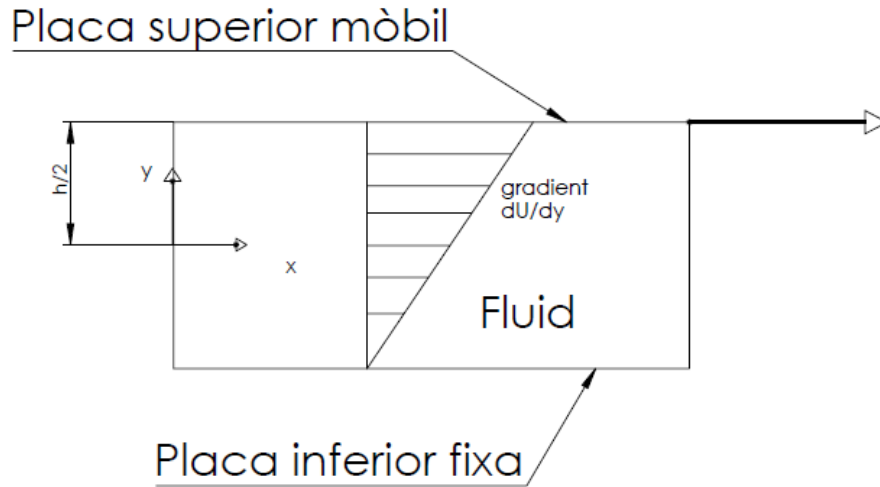


Figura 9.1: Esquema del cas 1

9.2.1 Desenvolupament físic

La placa superior es mourà en l'eix de les x de forma horitzontal així com es planteja a la figura 9.1. Si s'aplica a l'equació de continuïtat:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \rightarrow \frac{\partial u}{\partial x} = 0 \rightarrow u = u(y) \quad (9.1)$$

Equació del moment a l'eix de les x:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (9.2)$$

Si U es mou a velocitat constant damunt un pla i es negligeixen els gradients de pressions l'equació de Navier-Stokes se simplifica:

$$\frac{d^2 u}{dy^2} = 0 \rightarrow u(y) = C_1 y + C_2 \quad (9.3)$$

Per tant si s'aplica les condicions de contorn següents:

$$u = U \rightarrow y = \frac{h}{2} \quad (9.4)$$

$$u = 0 \rightarrow y = -\frac{h}{2} \quad (9.5)$$

Per tant la solució és:

$$u = \frac{U}{h}y + \frac{U}{2}, \quad \frac{h}{2} \leq y \leq \frac{h}{2} \quad (9.6)$$

Aquesta és la solució del cas *fluxe de Couette* amb una paret mòbil. On el perfil de velocitats és lineal. [1]

9.2.2 Resolució amb OpenFoam i Pre-processing

Per començar amb la resolució del problema amb Openfoam, s'afegiran els valors de $h=0.2$ m i els plans una longitud de 2m. La velocitat amb la que es desplaçarà el pla superior serà de 1m/s. A continuació s'explicarà en detall el procediment del pre-processing.

Abans de res cal crear una carpeta on es treballarà amb aquest cas. Per tant, s'ha d'obrir el terminal i una vegada obert s'introdueix el següent:

mkdir -p CFD/MMobil (9.7)

En treballar en Windows, també es pot fer de forma usual, una vegada es creada la carpeta que uneix Windows amb Linux explicat al capítol anterior.

Amb aquest ordre indicat al terminal, a continuació, es crea al directori principal una carpeta amb el nom de **CFD** i una altra a dintre **MMobil** que serà el cas que es solucionarà a continuació.

Una vegada indicada l'ordre anterior s'accedirà a dins el cas particular per tal de començar amb el pre-processing i dintre de **MMobil** es crearan tres carpetes: constant, system i 0.

- Dintre **constant** es trobaran tots els arxius que fan referència a les propietats físiques que són necessaries per al cas i també la descripció de la geometria de la malla.
- A **system** s'hi troben totes les ordres el que fa referència a solucionadors i paràmetres per solucionar el cas.
- **0** hi ha tots els arxius referents a les condicions inicials referents al problema.

Aquestes carpetes seran creades primer accedint dintre de la carpeta:

cd CFD/MMobil (9.8)

Amb l'ordre **cd** s'accedeix dintre d'una carpeta. Amb l'ordre **mkdir** es crearan les tres carpetes:

mkdir constant system 0 (9.9)

9.2.2.1 Creació de la geometria

El MMobil serà un prisma de 2x0.1x0.01. Es crearà aquesta malla amb aquestes dimensions. El que pot dur a confusió és que anteriorment s'ha esmentat que seria de dues dimensions però Openfoam treballa amb tres dimensions i a més amb un sistema Cartesià de coordenades.

Una vegada creades les carpetes anteriors, el que es farà és accedir a la carpeta **constant** on es crearà una altra carpeta que s'anomenarà **polyMesh**.

Per tant:

cd constant (9.10)

A continuació una vegada accedit a la carpeta **constant** es crearà **polyMesh**.

mkdir polyMesh (9.11)

Llavors es crearà dintre de **polyMesh** un arxiu **gedit**. És un editor de texts. Simplement ha d'obrir el programa **gedit** des del windows, introduir el codi desitjat i guardar-lo dintre de la carpeta corresponent, en aquest cas dintre de la carpeta **constant** creada anteriorment i dintre d'aquesta, una vegada dintre, dintre de **polyMesh** amb el nom corresponent de **blockMeshDict**.

El codi que fa referència a la geometria de malla i les seves característiques que s'implementa per generar la malla es comentat a continuació. El codi es pot trobar a l'**annex 1.1.1**.

- De les línies 1-15. Es la capçalera del programa on es pot veure la versió i el nom de la carpeta a la línia 13.
- A la línia 17, apareix la funció **convertToMeters** que en aquest cas és 0.1. Per tant vol dir que multiplicarà per 0.1 les distàncies en què és definiran vèrtexs. Per exemple si s'introdueix una longitud de 20, voldrà dir que són 2m per tant és com si fos en decímetres (dm).
- **vertices** fa referència als punts que calen destacar de la malla en aquest cas es tindrà un prisma per tant tindrem 8 punts i per donar la localització es fa de la següent forma *ex:(x,y,z)* (cada lletra fa referència a l'eix de coordenades corresponent) s'ha d'aplicar el coeficient indicat a la funció **convertToMeters**. A part els vèrtexs s'anomenen, el primer correspon al vèrtex 0, el segon correspon al vèrtex 1 i així successivament. Per generar un hexaedre, un arc, una cara... es fa referència al seu nombre per ordre.
- **blocks** defineix els diferents blocs que té la malla i es defineix així:

$$\text{hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)} \quad (9.12)$$

► **hex** fa referència a hexaedre.

► **(0 1 2 3 4 5 6 7)** aquests termes fan referència als vèrtexs de l'hexaedre que es vol crear. En aquest cas la malla és un simple hexaedre i s'han de definir els 8 vèrtexs es recorda que cada vèrtex té el seu propi nombre. Ex. 0 va referència al primer punt i així successivament.

► **(20 20 1)** fa referència al nombre de divisions que tindrà en cada eix local de l'hexaedre. En el cas de la imatge que hi ha a continuació seria $(x_1 x_2 x_3)$.

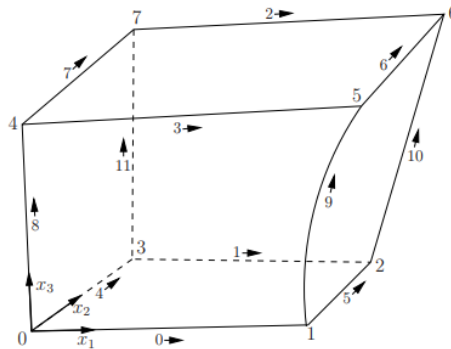


Figura 9.2: Bloc simple [4]

► **simpleGrading (1 1 1)**, és anomenat “Cell expansion ratios”, aquesta entrada permet per cada direcció del bloc graduar la malla o refinar-la a diferents direccions. La proporció és l'ample de la cel·la d'inici (δ_s) per tota l'aresta com es pot veure a la següent imatge:

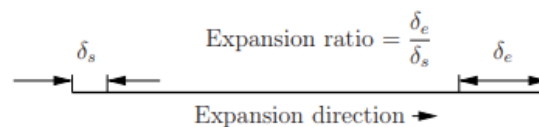


Figura 9.3: Cell expansion ratios [4]

La funció **simpleGrading** és la forma simplificada per indicar una expansió uniforme en les direccions del sistema de referència local.

- **Edges**, per defecte la forma d'unir dos vèrtexs sempre és una recta. Però per especificar que té la forma corba s'utilitza aquest apartat.
- **Boundary**, és conegut com el límit de la malla (contorn de la malla). Aquests límits de malla es divideixen en regions cada una amb un nom determinat per l'usuari. Aquestes regions més tard s'hi establiran les condicions de contorn. Se'n diferenciarien de tres tipus:
 - **wall**, aquesta funció s'aplica en casos on les condicions de contorn fan referència a un mur. Les cares de l'exemple anterior són **top** i **bottom**, que seran les plaques amb moviment relatiu.
 - **patch**, en aquest cas s'utilitza per denominar una regió on no conté informació extra de la geometria de la malla sinó informació per al problema. Les cares involucrades en aquest cas són **inlet** i **outlet** són les regions per on entra i surt el fluid.
 - **empty**, aquesta funció s'aplica als dos plans que determinen l'exemple en 2D i han de ser "buits" **frontAndBack** on es localitzen les parets que tanquen la malla.
- **mergePatchPairs**, és la funció que fusiona un conjunt de regions.

Aquest arxiu s'ha de guardar amb el nom de **BlockMeshDict** dintre de la carpeta **polyMesh**.

9.2.2.2 Condicions inicials i condicions de contorn

Una vegada s'ha creat l'arxiu que determina la geometria de la malla, el següent pas és determinar les condicions de contorn i les condicions inicials. Com bé diu el títol d'aquest subapartat les condicions inicials s'ubicaran a la carpeta 0 que fa referència al temps 0.

Es recorda que si amb el terminal és necessari tornar a un directori anterior, es fa de la següent manera:

```
cd ..
```

Dintre del directori **0** es crearan dos arxius **gedit** anomenats: **p** (que farà referència a la pressió) i un altre **U** que representarà la velocitat.

El codi que tindrà el primer arxiu que s'anomenarà **p** (fa referència a la pressió), es pot trobar a l'**annex 1.1.2**.

El segon arxiu a generar s'anomenarà U, fa referència a la velocitat es troba a l' **annex 1.1.3**.

Els dos arxius fan referència a les propietats físiques que té el cas en $t=0s$. El primer amb la pressió i el segon amb la velocitat. Els arxius segueixen el mateix ordre. Primerament es defineixen les unitats amb la funció **dimensions**. A continuació es defineixen el camp interior **internalField** a $t=0$, després un per un tots els contorns que s'han generat al fitxer **blockMeshDict**. A continuació s'explicarà de forma detallada cada funció.

- **dimensions**

Aquesta funció serveix per definir la magnitud amb la qual es vol treballar per atorgar-hi condicions de contorn.

Taula 9.1: Unitats bàsiques [4]

No.	Property	SI unitats
1	Mass	kilogram(kg)
2	Longitud	metres(m)
3	Temps	segons(s)
4	Temperatura	Kelvin(K)
5	Quantitat	mol(mol)
6	Corrent	ampere(A)
7	Intensitat de llum	candela(cd)

El nombre defineix la posició de dins la taula per definir la unitat que es vol tractar.

Per exemple: Si es vol definir la velocitat que té per unitats m/s per tant:

$$\text{dimensions (0 1 -1 0 0 0 0)} \quad (9.13)$$

En el cas de la pressió s'ha d'anar amb compte, ja que com es pot observar $[0 \ 2 \ -2 \ 0 \ 0 \ 0 \ 0]$ i la pressió no té m^2s^{-2} sinó que $Pa = kg^1m^{-1}s^{-2}$ però en aquest programa no es treballa amb pressió sinó que es treballa amb pressió cinemàtica $\frac{p}{\rho}$.

- **internalField**

Aquesta funció conté les dades internes dintre del camp intern en el moment inicial de la simulació, es pot iniciar a un valor aleatori i en funció del temps evolucionarà. Però en aquest cas un camp de pressions no és important, ja que es considera un fluid incompressible per tant s'establirà **uniform 0**.

- **boundaryField**

Amb aquesta funció s'inclouran les condicions de contorn de la unitat que s'està tractant per cada regió que s'ha determinat anteriorment a la malla.

En el cas de les regions **top** i **bottom** es vol simular que són plaques i com s'ha explicat en el problema, la placa de dalt es desplaçarà a $1m/s$ de forma uniforme i altre serà fixa. En les regions, s'haurà d'indicar **type**(tipus) i es vol un valor fixa per tant **fixedValue** i a continuació s'ha d'especificar si és **uniform** o **nonuniform** i a continuació el seu valor. En forma d'escalar o en forma de vector depenent de l'arxiu que s'està utilitzant. S'ha

de tenir en compte, és que el vector que s'introdueix fa referència a l'eix de coordenades global.

Exemple: Per donar valor a la velocitat de la placa de dalt que es mou a 1m/s, aquestes serien les següents funcions:

type fixedValue

value uniform (1 0 0)

Aquesta ordre es troba a la línia 25 i 26 de l'arxiu U. Simplement per donar la condició de contorn de velocitat de la placa fixa, com que no es mou, seria la mateixa ordre però indicant que la seva velocitat és (0 0 0), ja que no es mou en cap direcció.

Una altra funció que apareix en els codis anteriors, en les regions **top** i **bottom** en l'arxiu que s'especifica la pressió és **zeroGradient**, aquesta funció indica que el gradient de pressions en aquest cas serà 0 i que al llarg del temps pot variar.

També s'utilitza en el fitxer de velocitat, ja que el valor és desconegut i el que es desitja és que s'adapti a les condicions internes del camp. Encara que la velocitat és un camp vectorial s'indica exactament igual. S'indica de la següent forma:

type zeroGradient

En els casos de la pressió es suposa que serà zero, ja que no es vol crear una diferència de pressions i que el flux es desplaci a causa de l'increment de pressions, sinó que poder observar el fenomen que produeix desplaçar la placa. Per això les regions **inlet** i **outlet** s'establirà que té un valor de 0 (la pressió al ser un escalar no s'ha de definir vectorialment) :

type fixedValue

value uniform 0

Finalment, també es fa servir la funció **empty** i s'utilitza a la regió **frontAndBack**, aquesta funció fa que no contribueixi a la resolució física del problema.

type emty

9.2.2.3 Propietats físiques

Per solucionar aquest problema s'utilitzarà la funció **icoFoam** que s'utilitza per problemes que són transitoris, incompressibles, laminars i líquids Newtonians. L'única propietat que necessita és la viscositat cinemàtica. Per tant s'ha de crear un arxiu anomenat **transportProperties** dintre de la carpeta **constant**. Per accedir a constant des de la carpeta

MMobil únicament cal:

cd constant

I crear l'arxiu **gedit** i introduir el següent codi inclòs a l'**annex 1.1.4**:

Anteriorment s'han definit les dimensions de la viscositat cinemàtica (nu vol referir-se a ν) que les seves unitats són m^2/s . Aquesta propietat també s'utilitza per canviar el Reynolds en altres casos. El Reynolds és un número adimensional que indica si un flux és laminar o turbulent, ja que per solucionar-lo amb la funció **icoFoam** és necessari comprovar-ho. La fórmula per calcular el nombre de Reynolds és:

$$Re = \frac{d|U|}{\nu} \quad (9.14)$$

d, fa referència a la longitud característica i en aquest cas és l'altura $d=h=0.1m$ i la **U** és la velocitat de la placa per tant $1m/s$ i $\nu = 0.01 m^2/s$ perquè el nombre de Reynolds sigui 40. Es recorda que el valor de la viscositat cinemàtica és molt gran (l'aigua a $5^\circ C$ $1.5 \times 10^{-6} m^2/s$). Això és per no crear inestabilitats inicials i que les forces viscoses siguin majors que les forces referents a la inèrcia (flux laminar).

9.2.2.4 ControlDict

Aquest apartat tracta sobre com crear un arxiu per dominar el control del temps de lectura i escriptura de la solució. Aquesta informació és llegida del fitxer anomenat **controlDict**, es troba dins el directori de **system**. Partint del directori del cas, únicament cal executar la següent ordre:

cd system

Aquí dintre serà on es crearà el fitxer **controlDict** amb el codi inclòs a l'**annex 1.1.5.A** continuació s'explicaran les funcions utilitzades:

- **application**

En aquesta funció s'indicarà amb quina funció que solucionarà el problema. En aquest cas serà **icoFoam**, ja que com s'ha esmentat serà un cas transitori amb flux laminar i un fluid newtonià.

- **startFrom**

Indica a quin temps ha de començar. En aquest cas començarà amb el valor de l'especificació següent anomenada **startTime**.

- **startTime**

És el temps que comença la simulació. El moment de començar dependrà del nom

que tingui la carpeta. Es pot començar al 0.5 però la carpeta no pot començar en 0 ha de ser 0.5.

- **stopAt**

Indica el temps que ha d'haver acabat la simulació, llegeix de l'especificació **endTime**

- **endTime**

És el moment en què la simulació acaba. L'objectiu és arribar a l'estat estacionari encara que la funció **icoFoam** és transitòria, es considera estacionari quan no hi ha turbulència. Com a regla general el fluid ha de passar pel domini 10 vegades per arribar a ser laminar. Això implicaria un temps de finalització igual a 20. S'ha vist que 5 és suficient.

- **deltaT**

Aquest terme és el pas del temps per cada interval. Per aconseguir una bona precisió i estabilitat numèrica, quan s'executa **icoFoam** es requereix un nombre de Courant menor que 1. El nombre de Courant es defineix com:

$$Co = \frac{\delta t |U|}{\delta x} \quad (9.15)$$

On δ és el pas del temps per interval, U és la magnitud de velocitat a través de cel·la i δx és la longitud de cada cel·la. La velocitat varia en cada cel·la de la malla i s'ha d'assegurar que per tot el nombre de *Courant* es compleix que $Co < 1$ en totes les parts. S'agafa la velocitat més gran i de cel·la més petita. Per tant es calcularà la longitud de la cel·la(δx), n és el nombre de cel·les:

$$\delta x = \frac{h}{n} = \frac{0.2m}{20} = 0.01m \quad (9.16)$$

Per consegüent:

$$Co = \frac{\delta t U}{\delta x} \rightarrow \delta t = \frac{\delta x Co}{U} = \frac{0.01 * 1}{2} = 0.005s \quad (9.17)$$

- **writeControl**

L'usuari estableix el temps el qual s'escriuen els resultats. En aquest cas hi ha la funció **timeStep** que escriurà cada n períodes de temps, on n és especificat més a baix a **writeInterval**.

- **writeInterval**.

L'usuari decideix quants resultats es volen escriure. Si la simulació comença a $t=0s$ i acaba $t=5s$. Si es desitgen els resultats cada 0.1s, es divideix el temps que l'usuari desitja entre δt .

$$\text{writeInterval} = \frac{0.1}{0.005} = 20 \quad (9.18)$$

Per tant 20 és el valor que ha de tenir **writeInterval**.

9.2.2.5 fvSchemes i fvSolution

Dintre de la carpeta **system** s'hi han d'afegir dos arxius **gedit** que s'han d'anomenar **fvSchemes** i **fvSolution**. El primer especifica els esquemes de discretització del volum. L'altre conté especificacions dels "solucionadors", les toleràncies admissibles de les equacions lineals i controls d'algoritmes.

El codi que fa referència a **fvSchemes** es troba a l'**annex 1.1.6**. I el següent a **fvSolutions** a l'**annex 1.1.7**.

A versions anteriors les línies de codi que vénen a continuació, no calia posar-les però en aquesta és essencial, ja que sinó a l'hora de resoldre no et permet fer la simulació. [7]

```
pFinal                                     1
{                                           2
$P;                                       3
relTol 0;                                4
}                                         5
```

9.2.3 Creació de la malla

A la imatge següent es facilita un esquema per tenir clar el concepte de totes les carpetes i fitxers. Les carpetes apareixen en verd i en blau els diferents fitxers **gedit**.

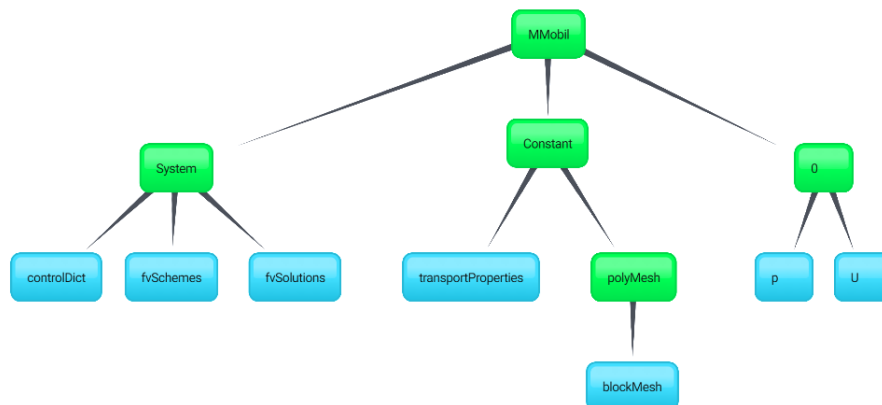


Figura 9.4: Carpetes i fitxers que ha de tenir el cas

Si no es té creat aquest esquema, no es podrà generar la malla. A continuació dintre el directori **MMobil** s'indicarà al terminal:

blockMesh

Si hi ha cap error, apareixerà al final del codi que produeix el terminal, la paraula **End** això significa que ha aconseguit crear la malla. En el cas que hi hagués algun tipus d'error apareixeria al terminal. Si es va dins la carpeta, **constant/polyMesh** haurà generat automàticament una diversitat de fitxers. Aquests fitxers contenen informació sobre les diferents regions, nombre de cel·les, nombre de cares, dels punts, etc.

A continuació per comprovar que no hi ha cap classe d'error a la malla s'ha d'executar la funció següent:

checkMesh

Aquesta funció conté informació sobre la qualitat de la malla. Per exemple factors com **Mesh nonorthogonality** or **Max skewness** són diferents indicadors de la qualitat de la malla. Si la malla és correcta, apareix el següent:

Mesh OK.

A continuació és hora de presentar el que serà el programa que ajudarà amb el post-processing en aquest cas s'utilitzarà per visualitzar la malla, s'anomena **ParaView** per executar-lo i veure la malla creada només cal escriure al terminal:

paraFoam -touch -builtin

Aquesta funció crearà un arxiu no temporal, amb el nom de la carpeta en aquest cas **MMobil.foam** dintre de la carpeta **MMobil**. Una vegada creat, l'usuari haurà d'obrir el programa paraView instal·lat a l'ordinador. A continuació dalt a l'esquerra apareixerà **file** i dintre open, si es desitja, es pot anar directament on apareix una carpeta i cercar l'arxiu **MMobil.foam** i obrir-lo.

Una vegada seguit el procediment anterior, per visualitzar la malla, s'ha de seguir el següent procediment:

1. Clicar damunt la pestanya **Pipeline Browser**.
2. A continuació clicar damunt l'ull per poder visualitzar.
3. Clicar damunt **Surface** i seleccionar **Wireframe**.

Aquests són els passos per visualitzar la malla. Aquí es deixa una imatge per obtenir una millor comprensió 9.5.

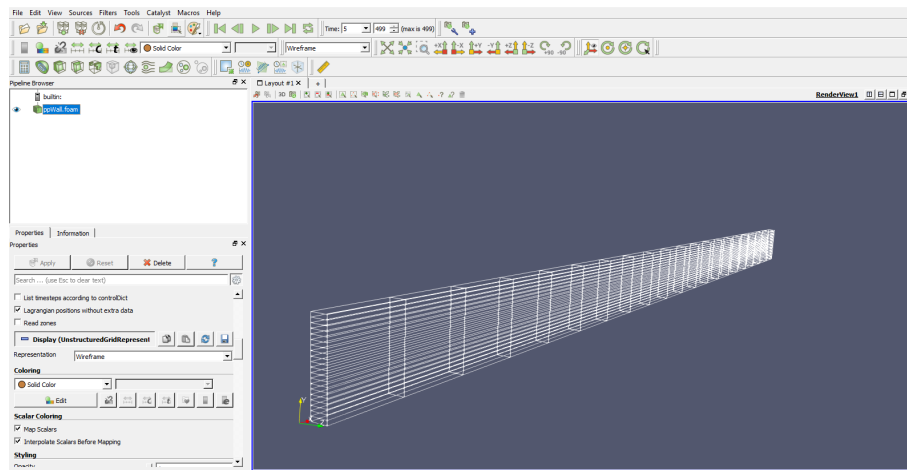


Figura 9.5: Malla del cas

També es poden visualitzar els diferents contorns **top**, **bottom**, **outlet**, **inlet**, etc. A la pestanya **properties** i seleccionant les cares que es volen visualitzar i clicant a **Apply**.

9.2.4 Executar la solució

Una vegada visualitzada la malla s'ha de tancar la finestra **ParaView**. Per poder procedir a els càlculs i que el programa OpenFoam treballi amb totes les condicions especificades a la carpeta MMobil, s'ha d'assegurar que s'està dintre de la carpeta MMobil, però no dintre d'una carpeta interna d'aquesta(no dintre System,Constant ni 0), si s'està dintre d'una d'aquestes escrivint al terminal **cd ..**, torna una carpeta anterior. Una vegada complint la condició anterior s'executa el solver en aquest cas **icoFoam** simplement escrivint al terminal:

icoFoam

Una vegada executada el terminal començarà a executar l'ordre i una vegada acabat es pot observar el temps d'execució, el màxim nombre de Courant. S'hauran creat diferents carpetes en funció del temps de l'interval en el qual fa referència a la combinació explicada anteriorment per **deltaT** i **writeInterval** definides anteriorment al fitxer **controlDict**.

9.2.5 Postprocessing i executació de la aplicació

Per poder començar amb el post-processing, s'executarà al terminal la mateixa funció que per veure la malla:

paraFoam -touch -builtin

A continuació es generarà un arxiu no temporal, MMobil.foam, dintre de la carpeta MMobil es tornarà a obrir el paraview. S'obrirà l'arxiu generat que s'ha subscrit a l'anterior que només es podria veure la imatge 9.6 i 9.7.

Aquests són els resultats esperats:

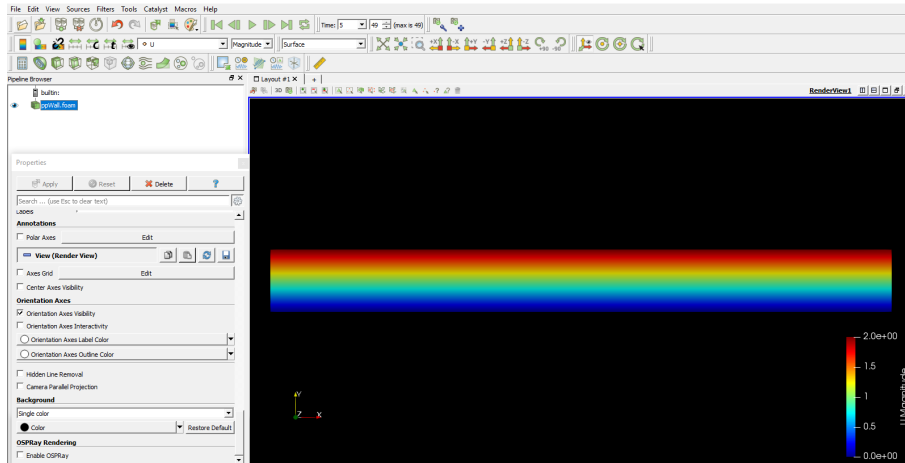


Figura 9.6: Camp de velocitats

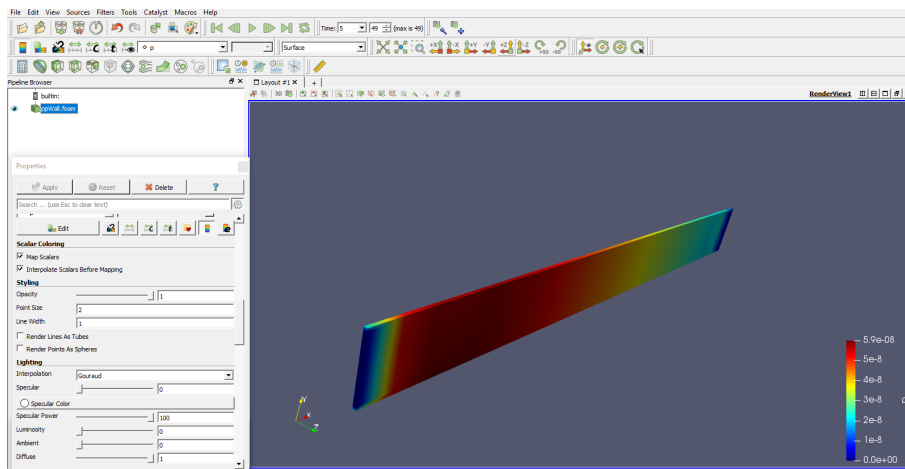


Figura 9.7: Camp de pressions

9.2.5.1 Visualització de resultats

Per poder observar aquests resultats:

1. Clicar damunt la pestanya **Pipeline Browser**.
2. A continuació clicar damunt l'ull per poder visualitzar.
3. Entrar dins la pestanya **Properties**.

4. Mirar que estiguin seleccionades **p** o **U**. Que són les magnituds que es volen visualitzar.
5. Clicar damunt **Apply**.
6. Canviar l'opció **SolidColor** per la unitat que es vol visualitzar (es troba a dalt de la pantalla). I clicar damunt **Apply**.
7. Per poder visualitzar com varia en funció dels graons de temps que s'han generat a la simulació, únicament cal, moures endavant i endarrere a la funció que posa **time**. És important situar-se a l'últim graó de temps per veure els resultats esperats de les equacions. En aquest cas a la funció time s'ha d'introduir l'últim valor **49**.
8. S'ha de rescalcar els valors dels colors. Clicant el botó indicat a la figura 1.7.
9. Es pot visualitzar una simulació animada, en funció del temps clicant damunt el ► (play).

9.2.5.2 Representació de variables

És possible obtenir una representació gràfica de diferents magnituds. En aquest cas s'explica com treure el gràfic de perfils de velocitats:

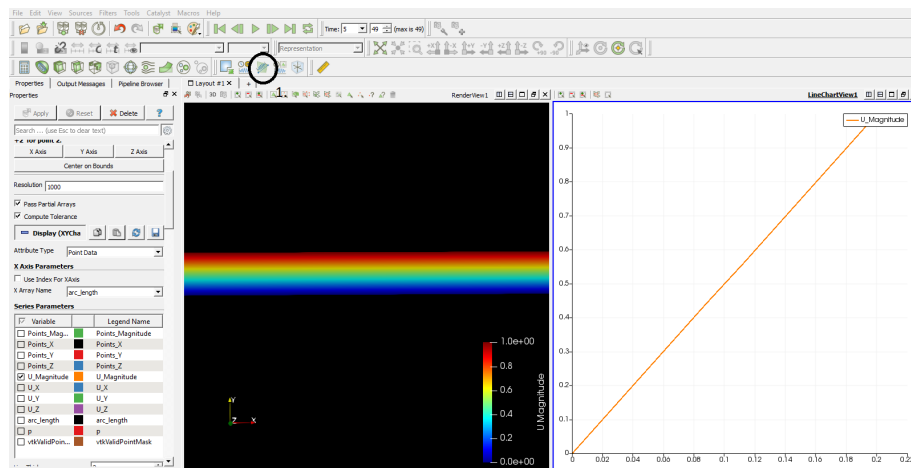


Figura 9.8: Generació del perfil de velocitat en funció de l'eix y

Per representar la variable en funció de l'altura com a la imatge anterior es pot clicar al nombre 1 que apareix a la figura 9.8, si no, també és possible d'una altra forma que és clicar **Filters** → **Alphabetical** → **Plot Over Line**. A continuació dins l'apartat de **Properties** seleccionar **Y axis** i clicar **Apply**, sortirà un gràfic de com varia. Es pot modificar l'eix de les abscises de forma manual donant valors a l'apartat de **Properties** on hi ha **Point1** i **Point2** també es pot fer de forma més visual arrossegant les dues boletes que generen una recta que apareixen a la malla, per variar la secció que es vol representar gràficament.

9.2.5.3 Representació de vectors

Hi ha diverses formes d'aconseguir el perfil de velocitats. Una s'ha explicat anteriorment fent la representació d'una variable. Per altre banda, es poden representar els vectors, com s'explica en aquest apartat.

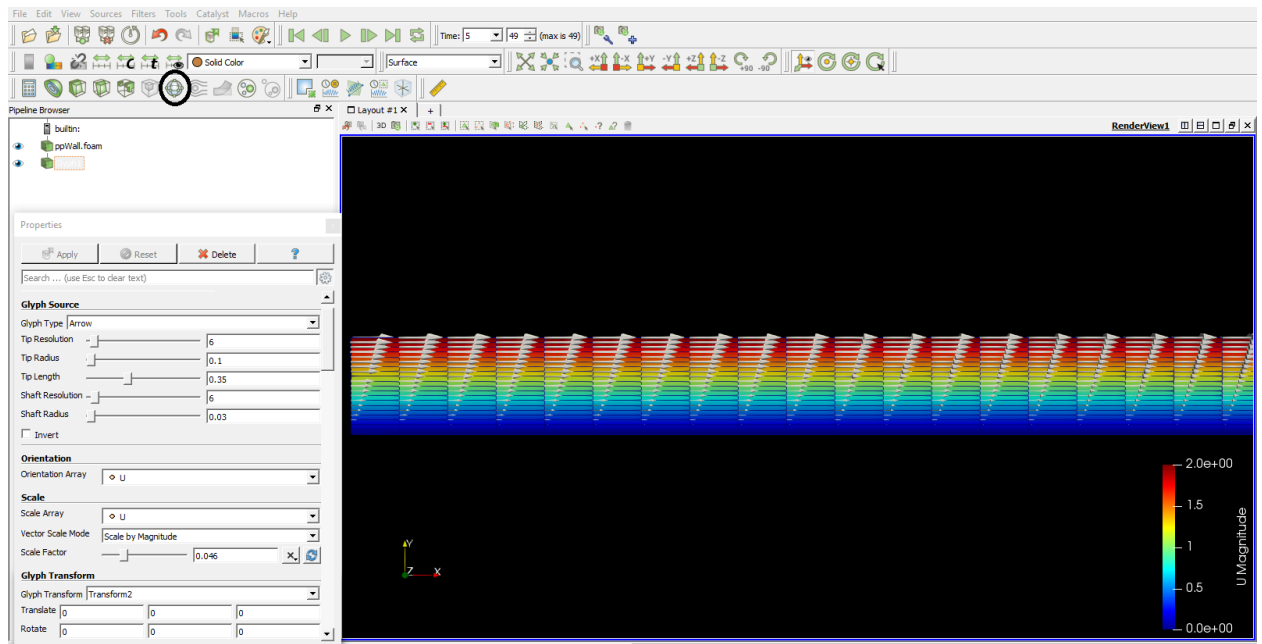


Figura 9.9: Generació del perfil de velocitats representant-ho amb vectors damunt la superfície.

1. Clicar a la icona que posa **Glyph** que és el seleccionat a la **Figura 9.9** o també es pot trobar d'una altra forma: **Filters** → **Alphabetical** → **Glyph**.
2. Anar a la pestanya **Properties**.
3. Seleccionar a l'apartat Glyph type **arrow**.
4. Escalars **p** i a vectors **U**.
5. Més a baix hi ha **Scale Mode** i s'ha de seleccionar **vector** i el factor d'escala 0.046.
6. Clicar **Apply**.
7. Si no es visualitza, entrar dins la pestanya **Pipeline Browser** i clicar damunt la icona que apareix un ull.

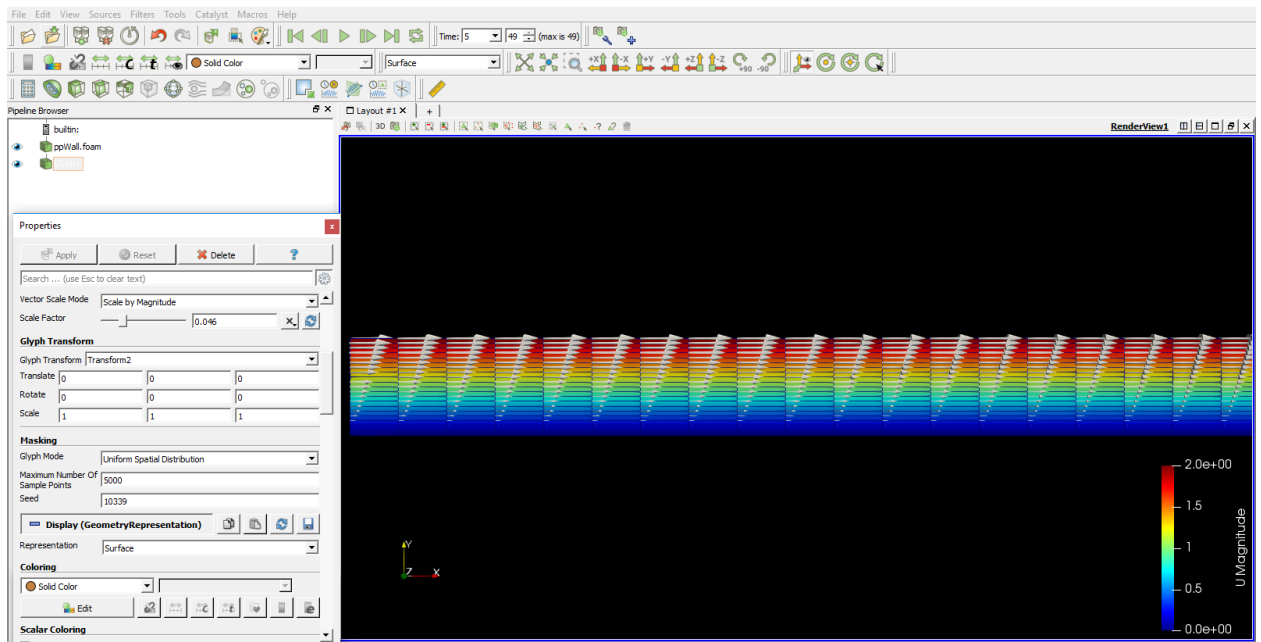


Figura 9.10: Al costat de la imatge hi ha els valors aconsellables per tenir una bona visualització.

9.3. Cas 2: plans paral·lels amb un gradient de pressió (plane-Poiseuille flow)

Aquest cas té exactament la mateixa geometria que l'anterior però a l'entrada tindrà una pressió (P_1) a l'entrada (regió inlet), una pressió a la sortida (P_2 , regió outlet). On $P_1 > P_2$ i a més fixar les dues plaques on no hi haurà cap mena de desplaçament. [1]

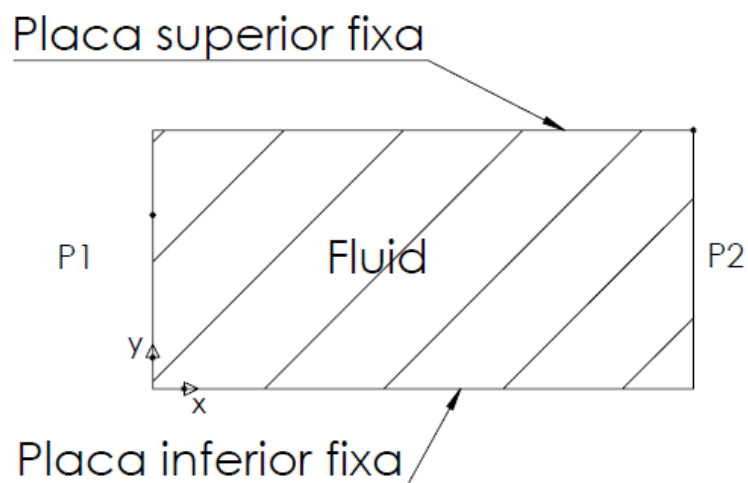


Figura 9.11: Representació gràfica del segon cas

9.3.1 Resolució analítica

En el segon cas, ni hi ha moviment entre plans però sí que hi ha un gradient de pressió a l'eix de les x . Utilitzant l'equació de continuïtat i simplificant-la:

$$u = u(y) \quad (9.19)$$

Si es simplifica l'equació del momentum, a l'eix de les x s'obté:

$$\mu \frac{d^2 u}{dy^2} = \frac{\partial p}{\partial x} \quad (9.20)$$

La variació de la pressió en els eixos y i z serà zero això vol dir que la pressió únicament dependrà de l'eix x , matemàticament s'expressarà de la següent manera:

$$p = p(x) \quad (9.21)$$

A conseqüència de l'equació **9.21** s'arriba a la següent conclusió $\frac{\partial p}{\partial x} = \text{constant} < 0$. És constant ja que si dues quantitats són iguals i la primera està en funció de y i l'altre està en funció de x , llavors han de ser iguals a la mateixa constant. Per consegüent es pot afirmar que la pressió serà lineal i al només dependre de x $\frac{dp}{dx}$. Molt més senzill de computar $\frac{P_2 - P_1}{l}$. Al realitzar les dues integrals el resultat és el següent:

$$u = \frac{1}{\mu} \frac{dp}{dx} \frac{y^2}{2} + C_1 y + C_2 \quad (9.22)$$

Si s'apliquen les condicions inicials següents:

$$u = 0 \rightarrow y = \frac{h}{2}$$

$$u = 0 \rightarrow y = -\frac{h}{2}$$

Per tant la solució:

$$u = \frac{dp}{dx} \frac{1}{2\mu} (y^2 - (\frac{h}{2})^2); \quad -\frac{h}{2} \leq y \leq \frac{h}{2} \quad (9.23)$$

Aquesta és la solució general pel plane-Poiseuille amb un gradient de pressió que té un perfil de velocitats parabòlic. A més el punt màxim de la velocitat màxima que tindrà el perfil ve determinada per la següent expressió:

$$u_{max} = -\frac{h^2}{8\mu} \frac{dp}{dx} \quad (9.24)$$

9.3.2 Preprocessing i execució del programa

Conservant les mateixes dades geomètriques, únicament afegint els valors de pressió i modificant els moviments de les plaques. Conservant les mateixes hipòtesis, llavors simplement s'ha de modificar el codi per adaptar aquest cas a l'altre.

Per la resolució la distància entre plaques serà de 0.1m la longitud d'aquestes serà de 2m. La pressió a la regió **inlet** $P_1=101325$ Pa i a la regió **outlet** $P_2=0$ Pa.

Primerament es crearan les carpetes adequades. En aquest cas en el directori de **CFD** es crearà una carpeta que es digui **MGrad**. Introduint dintre del terminal:

```
cd CFD
mkdir MGrad
cp -r MMobil/constant MGrad
cp -r MMobil/system MGrad
cp -r MMobil/0 MGrad
```

El que es fa en aquest cas simplement és copiar la carpeta sencera. A partir de la nova carpeta **MGrad** es faran les modificacions pertinents. També es pot fer dintre la carpeta de documents sense utilitzar les ordres des del Linux. Simplement copiant la carpeta i canviant el nom per **MGrad**, des de el Windows. Ja que aquesta carpeta està connectada entre Linux i Windows.

9.3.2.1 Condicions inicials i condicions de contorn

És necessari entrar dins el directori **0**, dintre del fitxer **U** i canviar el vector de velocitat de la regió **top** (0 0 0) d'aquesta manera la velocitat serà 0 en qualsevol direcció. El codi es pot trobar a l'**annex 1.1.8**.

El pròxim pas és adaptar la pressió en el cas proposat i s'ha de tenir en compte que aquest programari treballa amb la pressió dividida per la densitat ($\frac{p}{\rho}$). S'ha de tenir

en compte que es vol posar una pressió de 101325 Pa, però no es pot posar aquest valor directe, a causa de l'explicat anteriorment i es sap que la densitat és de $\rho = 1000 \text{ kg/m}^3$. Per tant el que s'ha de canviar és a la línia 36, que correspon a la regió **inlet** (entrada del fluid). A la sortida se suposarà una pressió de 0, que en el cas anterior ja estava indicat. El codi de la pressió es pot trobar a l'**annex 1.1.9**.

uniform 101.325

A continuació al tenir ja les altres carpetes i fitxers copiats ja no canvia res més del problema. Simplement s'ha de formar la malla, executant la funció **blockMesh**, una vegada generada la malla es pot comprovar amb la funció **checkMesh** i si la malla és correcte, es pot executar la funció per resoldre el cas escrivint **icoFoam**.

9.3.3 Postprocessing

A continuació s'ensenyaran els resultats d'aquest cas. El que s'ha de fer és entrar dins el visualitzador de **Paraview** amb l'ordre explicada al cas 1, però aquesta vegada des del directori **MGrad**, introduïnt la següent funció al terminal:

paraFoam -touch -builtin

I es visualitzaran seguint el mateix procediment que en el cas anterior. Es pot consultar en el punt **9.2.5**. Aquests són els resultats:

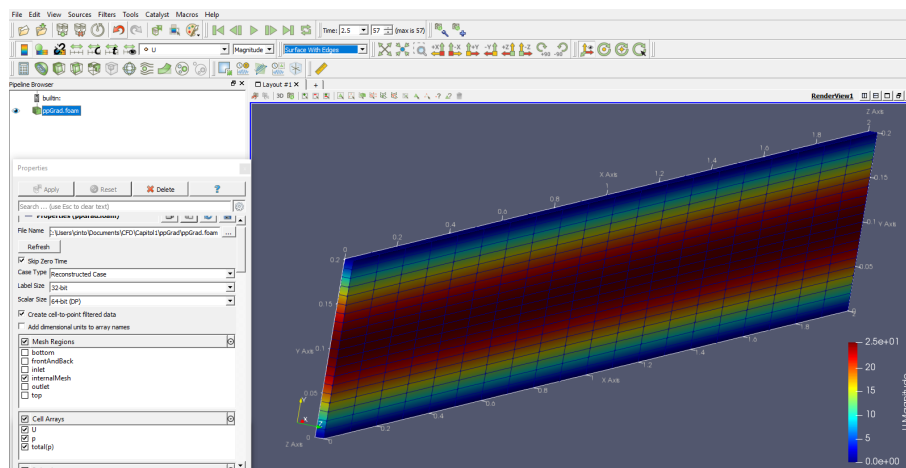


Figura 9.12: Representació del camp de velocitats del cas 2.

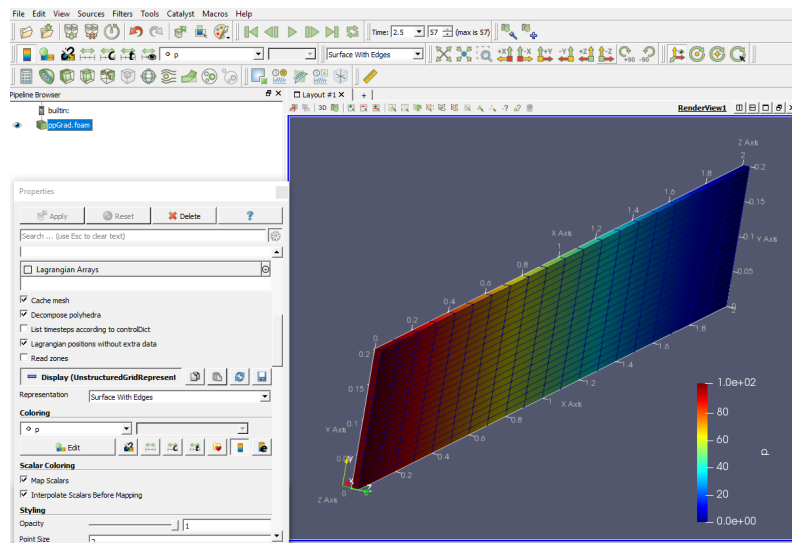


Figura 9.13: Representació del camp de pressions del cas 2.

Aquests resultats es poden obtenir seguint el mateix procediment a l'apartat 9.2.5.2.

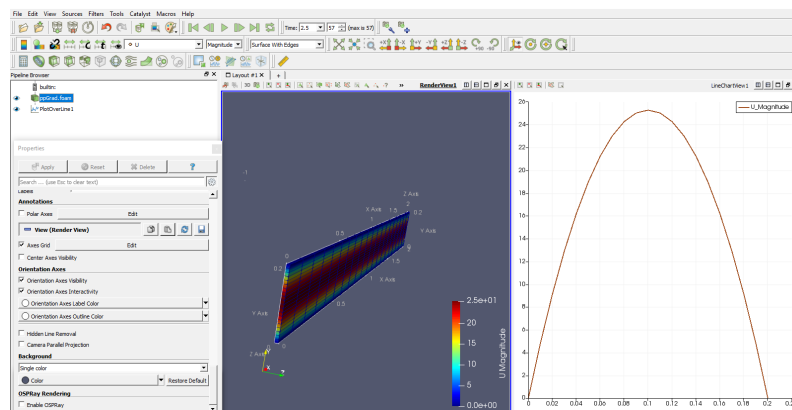


Figura 9.14: Representació de la velocitat en funció de la distància de l'eix Y.

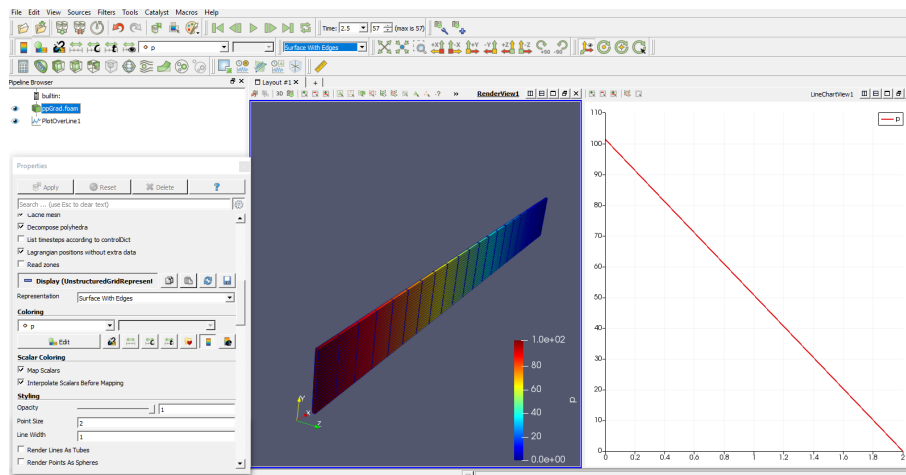


Figura 9.15: Representació de la pressió en funció de la distància de l'eix x.

Es pot observar que el perfil de velocitat en aquest cas té forma parabòlica i és uniforme al llarg de tot x . Que la pressió segueix una recta descendent a l'eix de les x tal com s'esperava, ja que una de les condicions era que a la regió **inlet** es tingués una pressió de 101325 Pa i a la regió **outlet** 0 Pa.

9.3.3.1 Càlcul del cabal

L'usuari pot executar funcions de post-processing després d'haver completat la simulació utilitzant funcions del **postProcess utility**. Hi ha diferents classes de funcions. Una funció que serveix d'ajuda és:

postProcess -help

Per calcular el cabal simplement dintre del directori cal utilitzar:

postProcess -func "flowRatePatch(name=outlet)"

Aquesta funció calcularà el cabal que passa per la regió que s'ha especificat a l'ordre. Les unitats vindran donades en funció del "solver" que s'hagi escollit. Si el solver és incompressible com per exemple **icoFoam**, el flux és volumètric per tant expressarà en m^3/s . En canvi si el "solver" utilitza un flux màssic per fer els càlculs, aquesta funció donarà les unitats en kg/s , sol ser a "solvers" compressibles. Simplement la forma de calcular-lo de forma analítica és [1] [3]:

$$Q = \int \vec{v} d\vec{S} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_0^z \frac{dp}{dx} \frac{1}{2\mu} (y^2 - (\frac{h}{2})^2) \cdot dz \cdot dy \quad (9.25)$$

La solució és:

$$Q = -\frac{dP}{dx} \frac{1}{\mu} \left(\frac{z \cdot h^3}{12} \right) \quad (9.26)$$

Per tant si es substitueix a l'equació 9.26 s'obté:

$$Q = -\frac{dP}{dx} \frac{1}{\mu} \left(\frac{z \cdot h^3}{12} \right) = -\frac{-101325}{2} \frac{1}{0.01} \left(\frac{0.01 \cdot 0.02^3}{12} \right) = 0.033775 \frac{m^3}{s} \quad (9.27)$$

On és el sumatori de tota una regió de la velocitat per cada diferencial de superfície d'aquesta regió. Una vegada el càlcul s'hagi acabat d'executar:

```

Time = 2.46
Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=inlet) write:
  sum(inlet) of phi = -0.0338637

Time = 2.5
Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=inlet) write:
  sum(inlet) of phi = -0.0338689

End

```

Figura 9.16: Finalització del procediment

A la imatge apareix el final de la simulació s'indica el temps i el cabal que passa per la regió. En aquest cas és negatiu, ja que va en sentit contrari a la normal del pla. Les unitats són en sistema internacional i apareixen amb m^3/s . En el cas que interressi una altra regió simplement s'ha de repetir el mateix procés indicat però indicant la regió desitjada. Si es fa aquest mateix càlcul a la regió **outlet**, sortirà el mateix valor però positiu, ja que la normal del pla va en el mateix sentit al qual flueix el cabal. Es pot observar que hi ha un error d'un 0.28%. Si la malla fos una mica més fina, s'ajustaria millor el resultat.

Si es desitja una major precisió, únicament cal fer la malla més fina. Dintre de l'arxiu **blockMeshDict** s'ha canviat el mallat de l'hexaedre, el nombre de divisions per hexaedre.

```

blocks
(
hex (0 1 2 3 4 5 6 7) (40 40 1) simpleGrading (1 1 1)
);

```

A part s'ha de canviar el **deltaT** seguint l'explicat a l'apartat 9.2.2.4 el nombre de Courant no pot ser major que 1 per tant: **deltaT**=0.0002, per mantenir el mateix nombre d'arxius escrits, la funció **writeinterval**=500. El temps de compilació augmentarà bastant, però el resultat és molt més acurat. En aquest cas la solució és $0.03374 m^3/s$ amb un error del 0.09%.

9.3.3.2 Càlcul de la pressió total

La pressió total és definida per $\frac{P}{\rho} + \frac{|U|^2}{2}$ per un flux incompressible amb pressió cinemàtica. La funció que s'ha d'utilitzar s'anomena **totalPressureIncompressible**. Perquè la funció s'executi amb èxit hi ha d'haver creada les carpetes de **p** i **U**. Hi ha dues opcions, una és carregar les carpetes com a arguments separats.

postProcess -func "totalPressureIncompressible(p,U)"

Alternativament, l'usuari pot carregar una llista separada d'espais separats mitjançant, utilitzant la funció següent:

postProcess -fields "(p U)" -func totalPressureIncompressible

Les dues opcions són vàlides, ja que funcionen de manera efectiva perquè les dades de pressió i velocitat estan directament als fitxers **p** i **U**. A continuació si s'executa el programa **paraView** amb l'ordre al terminal de **paraFoam**, apareixerà una nova magnitud per representar **p (total)**:

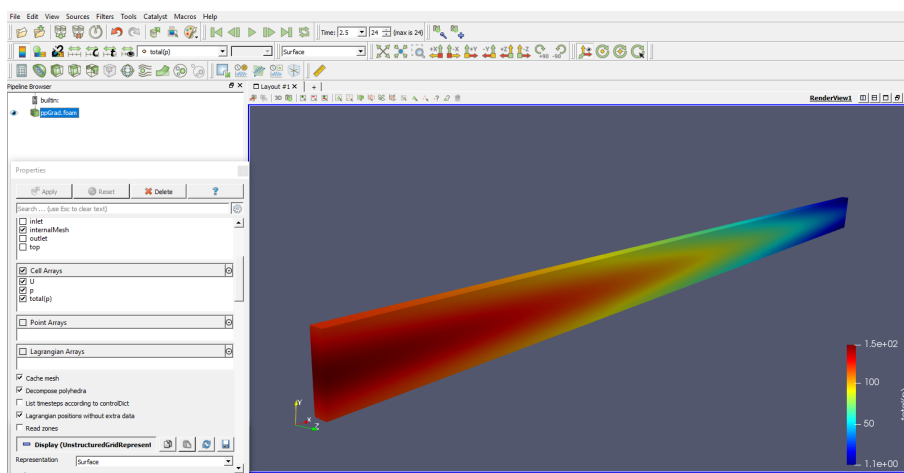


Figura 9.17: Visualització amb forma de camp de la pressió total

Es pot veure com es defineixen els perfils de velocitat. En ser una magnitud també es pot veure en forma de vector pel paraView com s'ha explicat a l'apartat **9.2.5**.

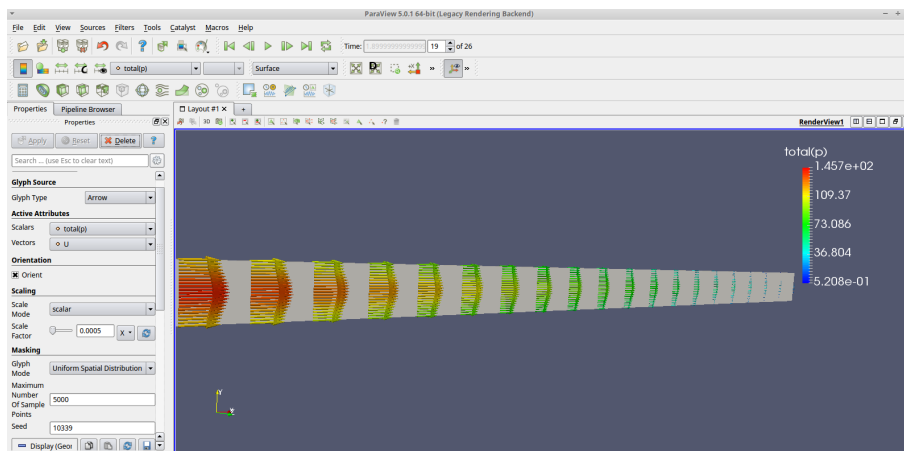


Figura 9.18: Visualització amb forma de vectors de la pressió total, $\frac{P}{\rho} + \frac{|U|^2}{2}$

9.3.3.3 Càlcul de l'esforç tangencial

Un altre exemple una mica més complex és el càlcul de l'esforç tangencial a la placa. A versions anteriors simplement era utilitzar la següent funció:

```
postProcess -fields "(p U)-func wallShearStress
```

Però ara és necessari utilitzar el solver **simpleFoam**. Ja que si no, no es podrà solucionar i sortirà un error dient: “Unable to find turbulence model in the database.”

Per tant es crearà un model turbulent. Abans de res, recordar que el solucionador **simpleFoam**, és un solucionador estacionari, això vol dir que els valors de temps, no fan referència a temps real sinó a temps d'iteració. Les hipòtesis d'aquest solucionador són:

- Incompressible.
- Laminar o turbulent.
- Isotèrmic.

Apareixen noves variables a tenir en compte que es farà una petita explicació de cada propietat.

- k , és l'energia cinemàtica de la turbulència $k = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2})$. Amb unitats $\frac{m^2}{s^2}$.
- ϵ , és una ràtio de la dissipació de l'energia cinemàtica de la turbulència. Unitats $\frac{m^2}{s^3}$.
- ν , en els fitxers escrit com a nut és la viscositat cinemàtica. $\frac{m^2}{s}$.
- nutilda, fa referència a la viscositat cinemàtica turbulenta. $\frac{m^2}{s}$.

En aquest cas es crearà una carpeta amb el nom **MGradShearStress** independent com si fos un altre exercici. Ha de tenir el següent esquema:

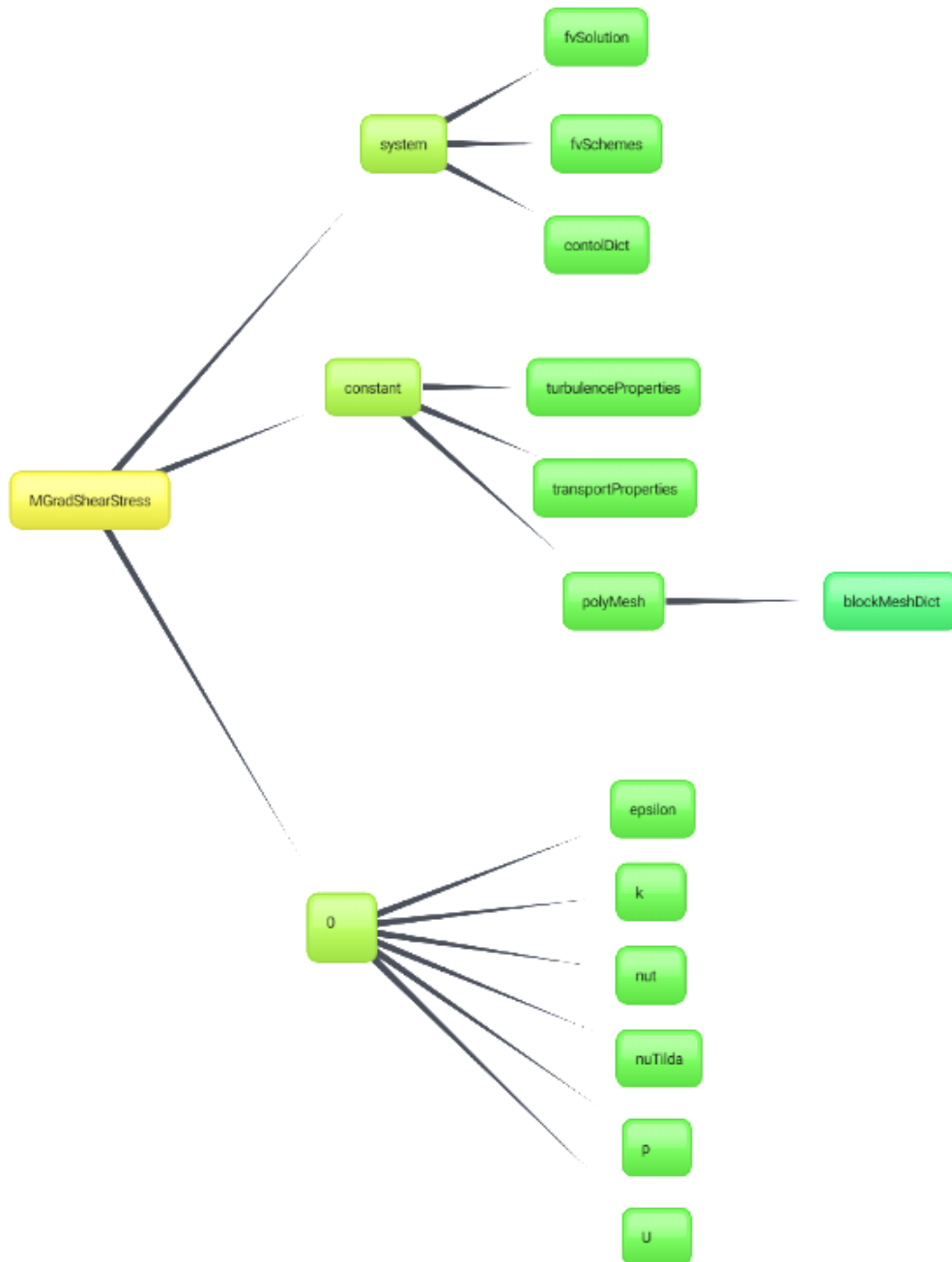


Figura 9.19: Esquema de fitxers del model Turbulent

El codi ve inclòs als **annexos 1.1.10**

Una vegada és creada la malla, arxius i fitxers necessaris, seguint l'esquema anterior amb el codi inclòs a l'annex anterior amb la funció **blockMesh**, a continuació s'hi ha d'introduir al terminal **simpleFoam**, és el solver amb resolució constant per a un flux turbulent incompressible, utilitzant l'algorisme SIMPLE.

Una vegada calculat, es visualitzarà al paraView escrivint al terminal **parafoam - touch -builtin** i aquests seran els resultats:

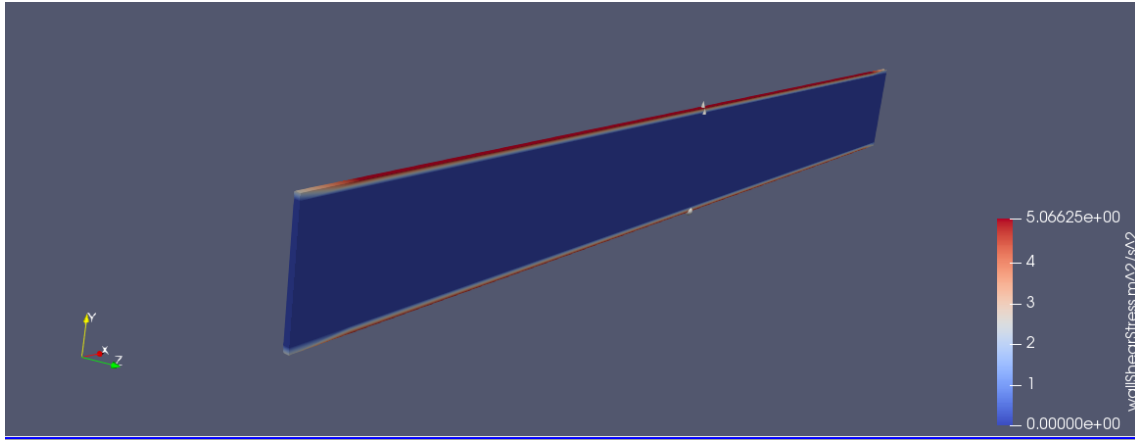


Figura 9.20: Esforços tangencials

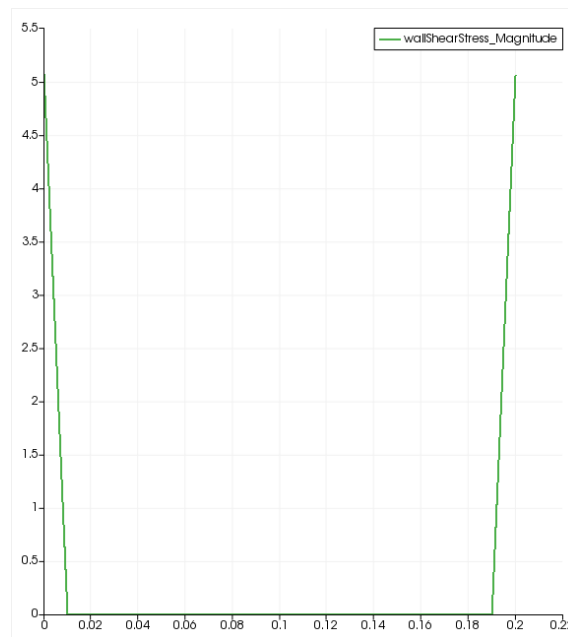


Figura 9.21: Variació dels esforços tangencials a l'eix de les y

Matemàticament, l'esforç tangencial a la regió **top** determinada anteriorment és:

$$\tau = \tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \Big|_{y=\frac{h}{2}} \approx \mu \left(\frac{\partial u}{\partial y} \right) \Big|_{y=\frac{h}{2}} \quad (9.28)$$

Si es combina amb l'equació 1.23 el resultat és:

$$\tau = -\frac{dp}{dx} \frac{h}{2} = -\frac{P_1 - P_2}{l} \frac{h}{2} = -\frac{101325 - 0}{2} \frac{0.2}{2} = -5066.25 Pa \quad (9.29)$$

Es faria exactament el mateix procediment per la regió **bottom** però aquesta vegada la condició de contorn varia $y = -\frac{h}{2}$. S'ha de tenir en compte que passa el mateix que en la pressió, el resultat proporcionat per el programa es donat en $\frac{\tau}{\rho}$, si es vol el resultat amb Pa. S'ha de multiplicar per la densitat en aquest cas $\rho = 1000 \text{ kg/m}^3$ per tant és 5066.25 Pa. En aquest cas coincideixen les solucions.

9.4. Cas 3: Plans paral·lels amb moviment ambdós amb la mateixa direcció però en sentit contrari

En aquest cas seguirà sent la mateixa geometria que els apartats anteriors, però en aquest cas, el que es modificarà serà el moviment. On apareixerà a les dues plaques amb la mateixa direcció però en sentit contrari i mateix mòdul.

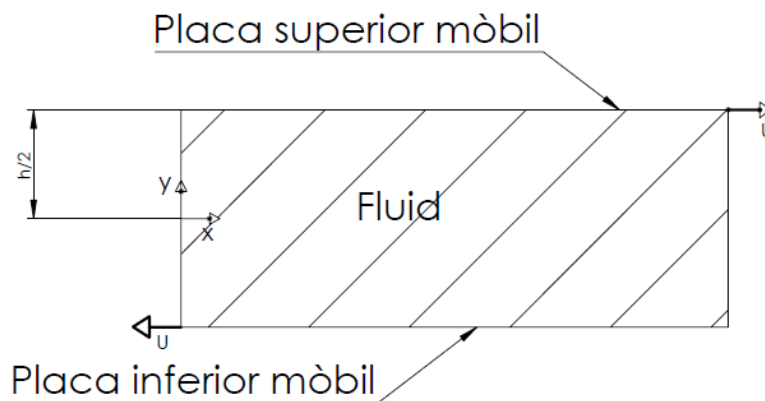


Figura 9.22: Dues plaques paral·leles mòbils amb la mateixa direcció i mòdul però diferent sentit.

9.4.1 Procediment físic

En tenir la mateixa geometria i les mateixes hipòtesis, únicament canvien les condicions de contorn per tant partint de l'equació 9.3 i amb les condicions de contorn següents:

$$u = U \rightarrow y = \frac{h}{2} \quad (9.30)$$

$$u = -U \rightarrow y = -\frac{h}{2} \quad (9.31)$$

Per tant si s'apliquen queda la següent equació:

$$u = \frac{2U}{h}y \quad (9.32)$$

Aquesta serà la recta que seguirà mentre es desplacin les dues plaques.

9.4.2 Pre-processing

En aquest apartat el que es farà és utilitzar la mateixa geometria anterior, les mateixes propietats del fluid únicament adaptant-les al problema proposat i a més crear una malla més fina per tal d'obtenir resultats més acurats.

Per tant el primer pas és crear una carpeta amb el nom **MMobils** amb els arxius corresponents i seguint el mateix esquema proporcionat anteriorment a l'apartat **9.4**. També es pot copiar la carpeta que ja segueix aquest esquema i modificar els diferents arxius corresponents, la forma de copiar les carpetes partint del terminal està explicada al **cas 2**. Els arxius a modificar lògicament seran **blockMeshDict** dintre de la carpeta **system** i també dintre de la carpeta **0** modificar el moviment de les plaques dintre de l'arxiu de **U**.

9.4.2.1 Refinament de la malla

Per poder refinar la malla se genererà dos hexaedres. El codi es pot trobar a l'**annex 1.1.11**. S'adjunta la figura 9.23, per poder veure de forma més gràfica l'ordre dels vertexs i com s'anomenen. A més s'indiquen les diferents regions i el seu nom. Les regions que apareixen són **inlet**, **outlet**, **back** i **front**.

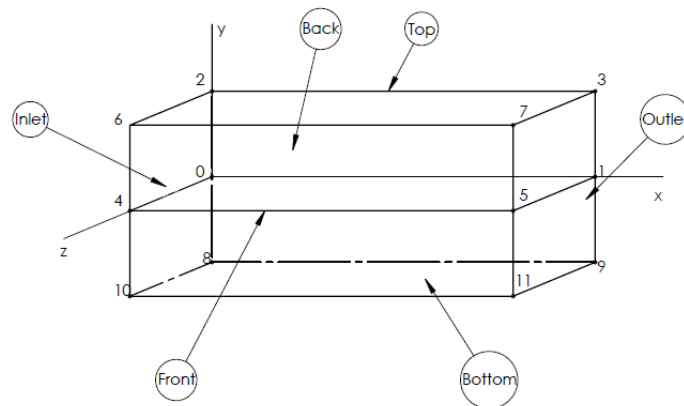


Figura 9.23: Esquema de la nova malla

S'han seccionat i s'ha creat la malla amb dos hexaedres distints. Això fa que si es conserven el nombre de cel·les per cada un dels hexaedres es tindrà el doble de cel·les que en el cas anterior. Però no només és important obtenir més seccions, sinó que també és important una graduació d'aquesta. Aquesta graduació proporciona aconseguir en zones molt concretes o d'especial interès una major precisió i no haver d'utilitzar malles excusivament fines.

En aquest cas s'utilitzen els factors de graduació de la malla indicats a les línies 37 i 38 del fitxer **blockMeshDict**. En el primer hexaedre (hexaedra superior), s'implementarà una graduació en l'eix de les y (seguint els eixos locals de l'hexaedre superior) i la relació d'altura entre la primera cel·la serà deu vegades més gran que la darrera cel·la. En canvi la línia 37 que faria referència el segon hexaedre(inferior). Al voler una expansió en sentit contrari ja que els eixos locals són idèntics a l'hexaedra superior i voler exactament la mateixa graduació. S'ha d'aplicar la inversa de la graduació del primer hexaedre.

9.4.2.2 Aplicació de les condicions de contorn de la velocitat

Simplement s'ha de canviar el moviment de les regions. En aquest cas s'ha de canviar la velocitat de la regió **bottom**. Com es pot veure implementat en el codi següent:

Com es pot observar en el codi a l'**annex 1.1.12**, s'ha de substituir el valor per -1 això farà que la regió se desplaçarà a -1 m/s s'ha de recordar que fa referència a l'eixos globals.

Una vegada s'han modificat els distints fitxers es pot procedir a executar la funció **blockMesh** per generar la malla. Una vegada generada tots els arxius estan creats, es pot visualitzar la malla utilitzant la funció **paraFoam** i es selecciona l'opció **wireframe**. El resultat ha de ser el següent:

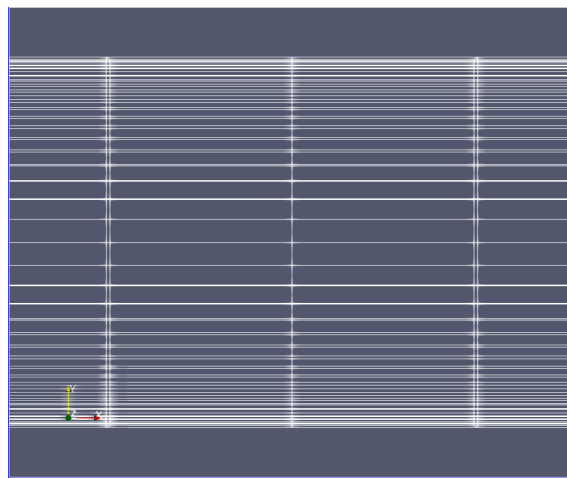


Figura 9.24: Nova malla

9.4.3 Postprocessing

Una vegada creada la malla es pot procedir a utilitzar el "solver". Sols caldrà introduir al terminal:

icoFoam

Una vegada ha acabat de calcular tota la informació, s'utilitzarà dintre del terminal l'ordre per visualitzar els resultats amb el programa **paraView**:

paraFoam – touch – builtin

En aquest cas i com s'ha explicat a l'apartat del post-processing es visualitzarà el perfil de velocitats i el camp de velocitats tal com s'indica als apartats 9.2.5.1 i 9.2.5.3. Es vol recordar que aquestes imatges s'han fet a l'últim interval de temps. Aquests seran els resultats:

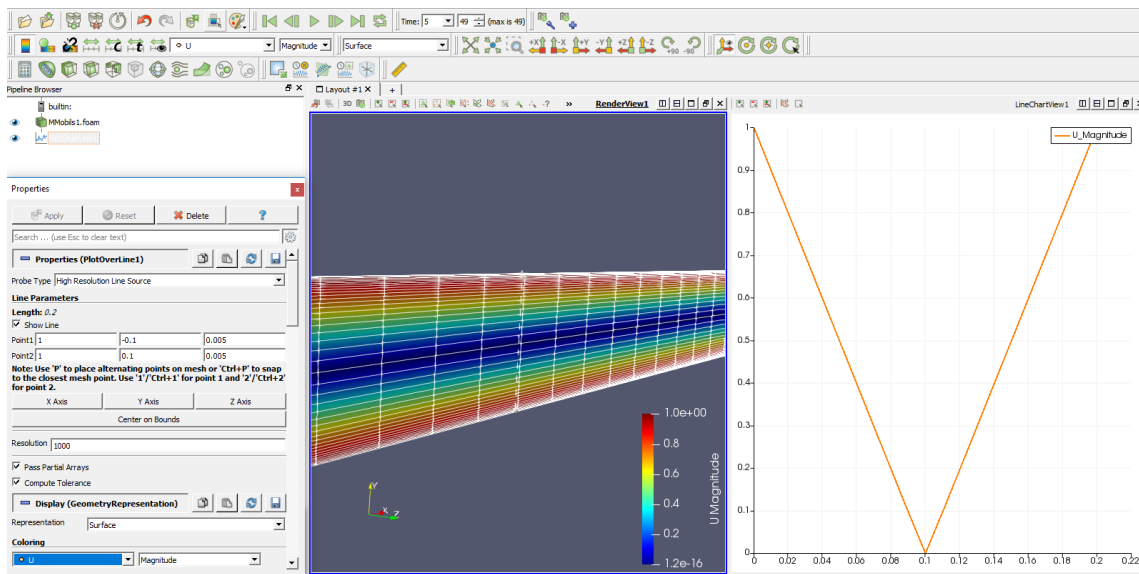
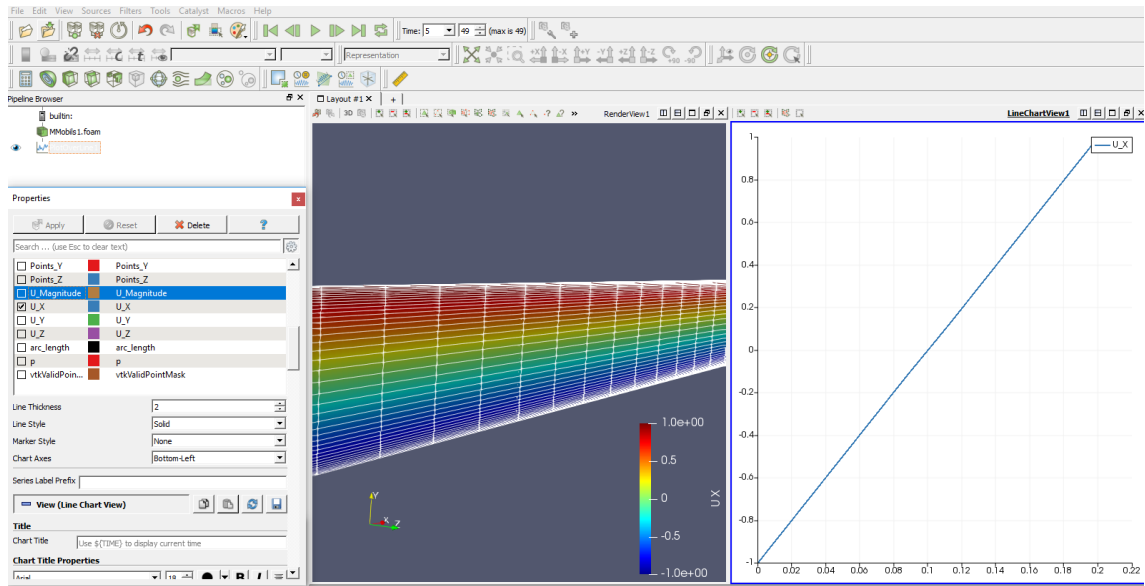


Figura 9.25: Visualització del perfil de velocitats amb valor absolut

S'ha de tenir en compte que aquesta visualització de la velocitat és en valor absolut que realment si es volgués saber la seva direcció, es podria fer ús de l'apartat 9.2.5.2 o també representar el valor de la variable U_X . Com s'ha fet en aquest cas.

Figura 9.26: Visualització de la variable U_x

9.4.3.1 Càlcul de cabal

El càlcul de cabal en aquest cas ha de ser 0, ja que és una recta que passa per 0. Es comprovarà matemàticament:

$$Q = \int \vec{v} d\vec{S} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_0^z \frac{2U}{h} \cdot y \cdot dz \cdot dy = 0. \quad (9.33)$$

Utilitzant el mateix procediment a l'apartat 9.3.3.1. El resultat és el següent:

```
Time = 5
Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=outlet) write:
  sum(outlet) of phi = 2.5411e-20
End
```

Figura 9.27: Finalització del procediment

Es pot veure que el resultat és de $2.5411 \cdot 10^{-20} m^3/s$ és un nombre molt petit que s'aproxima a 0.

9.4.3.2 Stremlines

Streamlines és una funció que disposa el programa, per crear línies mitjançant un filtre anomenat **StreamTracer**. És una bona opció per la representació del flux i es pot elegir quin tipus de magnitud desitja ser representada.

Abans de res és convenient:

- Canviar el color de fons de pantalla, perquè és possible que es pugui generar el color blanc i no es pogués visualitzar de forma correcta, posar un fons de pantalla de color obscur.
- A més, és convenient que esborrin dintre de la pestanya **Pipeline Browser** tots els arxius, menys per descomptat l'arxiu **MMobils.OpenFOAM**

Els passos a seguir són:

- Accedir als següents apartats **Filters** → **Alphabetical** → **StreamTracer**
- Accedir **Properties**. Clicar el botó **Y axis** i seleccionar la variable que es vol representar.
- A continuació cal baixar el nombre de "Resolution" a 50 així es podran diferenciar, si no pareixerà un sòlid.
- Tancar l'ull de l'arxiu **MMobils1.foam**, que està situat a la regió pipeline browser.
- Clicar el botó **Apply**.

El resultat hauria de ser com aquest:

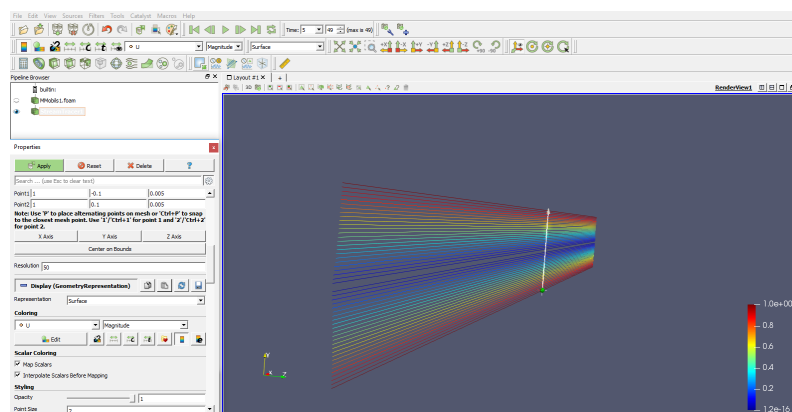


Figura 9.28: Visualització de StreamLines

10. Capítol 2: Estudi d'un flux per una canonada

En aquest cas s'estudiarà diferents casos, com un fluid circula per una canonada. Es posaran diferents casos. El primer serà una canonada simple, l'altre serà una canonada anular, amb una diferència i un cas final on rota el cilindre intern.

10.1. Cas 1: Flux d'una canonada amb un gradient de pressions.

Aquest cas consistirà en una canonada que hi circula un fluid on es desplaça a causa d'un gradient de pressions. Es presenta una imatge per entendre millor l'esquema. Aquest primer cas ha estat extret d'una altra guia però adaptant-lo a la versió 6 d'OpenFoam [1].

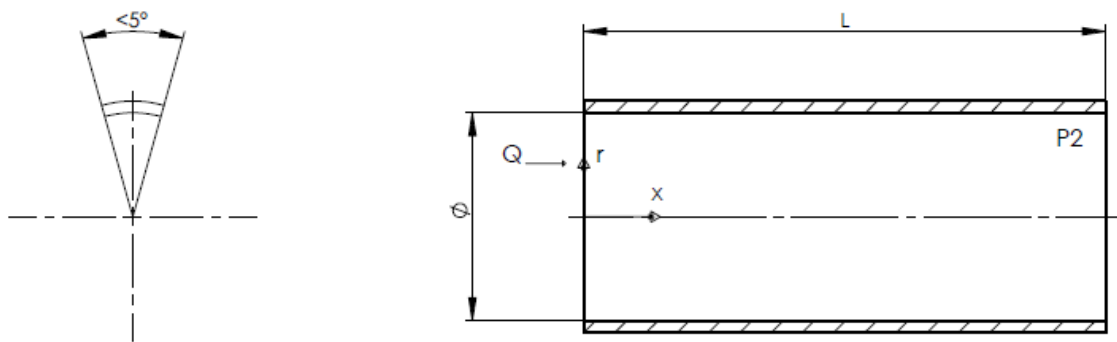


Figura 10.1: Esquema del cas

Com es pot observar s'ha agafat una secció d'angle, ja que com s'explicarà més endavant la velocitat en funció de l'angle serà 0. A continuació es solucionarà el cas de forma analítica. De longitud mesura 180 cm mentre que de diàmetre mesura 8 mm i passa un cabal de $25\text{cm}^3/\text{s}$.

A continuació es faran les següents hipòtesis:

- Flux incompressible.
- Flux laminar.
- Flux newtonià.

- Règim transitori.
- No lliscament entre fluid i canonada.
- No es tindran en compte les forces gravitatòries.
- Simetria cilíndrica $\frac{\partial}{\partial \theta} = 0$.

10.1.1 Resolució analítica

A una distància en concret, les capes límit s'incorporen i el fluid esdevé totalment viscos i la tensió tangencial es converteix en constant i la pressió varia linealment. Per resoldre de forma analítica aquest problema s'utilitzaran coordenades cilíndriques, (r, θ, z) r apunta cap a la paret de la canonada, és paral·lela a l'eix de la canonada. A una distància L_e , les capes límit s'incorporen i el fluid és totalment viscos.

Utilitzant l'equació de continuïtat en coordenades cilíndriques :

$$\frac{1}{r} \frac{\partial}{\partial r}(rv_r) + \frac{1}{r} \frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z} = 0 \rightarrow \frac{\partial v_z}{\partial z} = 0 \quad (10.1)$$

Considerant que l'única velocitat que no té component 0 es v_z , el significat d'aquesta equació ve a dir que a mesura que es desplaça per l'eix de la z la velocitat en z no varia una vegada $z > L_e$. Utilitzant l'equació del moment amb coordenades cilíndriques en la coordenada de l'eix z :

$$\rho \left(\frac{\partial v_z}{\partial t} + v_r \frac{\partial v_z}{\partial r} + v_\theta \frac{1}{r} \frac{\partial v_z}{\partial \theta} + v_z \frac{\partial v_z}{\partial z} \right) = - \frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 v_z}{\partial r^2} + \frac{1}{r} \frac{\partial v_z}{\partial r} + \frac{1}{r^2} \frac{\partial^2 v_z}{\partial \theta^2} + \frac{\partial^2 v_z}{\partial z^2} \right) \quad (10.2)$$

Simplement les altres equacions de moment indicaran que la pressió no variarà en direcció r i en direcció θ . A partir d'aquí utilitzant les condicions de no transitori, i la condició donada per l'equació de la continuïtat, sabent que $v_r = 0$, $v_\theta = 0$ i aplicant les hipòtesis se simplifica l'equació i utilitzant la llei de la cadena inversa, queda l'expressió següent:

$$0 = - \frac{\partial p}{\partial z} + \mu \frac{\partial^2 v_z}{\partial r^2} + \mu \frac{1}{r} \frac{\partial v_z}{\partial r} = - \frac{\partial p}{\partial z} \mu \frac{d}{dr} \left(r \frac{dv_z}{dr} \right) \quad (10.3)$$

Integrant dues vegades s'obté la funció de la velocitat, a part la pressió només depèn de z i tindrà forma lineal:

$$v_x = \frac{\Delta p}{\Delta z} \frac{r^2}{4\mu} + C_o \ln(r) + C_1 \quad (10.4)$$

Les condicions de contorn són les següents:

$$r = 0 \rightarrow v_z = V_{max} \quad (10.5)$$

$$r = R \rightarrow v_z = 0 \quad (10.6)$$

Aplicant les condicions de contorn s'obté la següent funció:

$$v_x = \frac{\Delta p}{\Delta z} \frac{1}{4\mu} (r^2 - R^2) \quad (10.7)$$

Sabent que la velocitat màxima serà:

$$v_{max} = -\frac{\Delta p}{\Delta z} \frac{1}{4\mu} R^2 \quad (10.8)$$

Realment la velocitat no es negativa, ja que normalment aquest valor $\frac{\Delta p}{\Delta z}$ és negatiu.

Es farà una comprovació per determinar si el flux és laminar. Sabent que $\rho = 900 \text{ kg/m}^3$ i que la viscositat cinemàtica és $\mu = 0.03 \text{ Pa} \cdot \text{s}$. [1]

10.1.2 Resolució amb OpenFoam

Primerament caldrà crear una carpeta dintre de la carpeta **CFD** creada anteriorment, generar els fitxers corresponent:

```
cd CFD                                1
mkdir pipe                             2
cd pipe                                3
mkdir constant 0 system                4
```

L'esquema de fitxers i carpetes serà el següent:

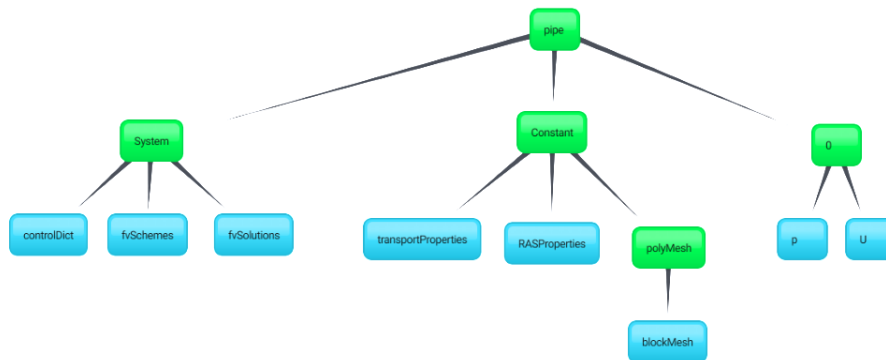


Figura 10.2: Esquema de fitxers i carpetes

En blau apareixen els fitxers i en verd les carpetes.

10.1.2.1 Constant

10.1.2.1.1 Generació de la malla

Dintre de la carpeta **constant** i dintre **polyMesh** s'ha de generar la malla en aquest cas s'utilitzarà una secció d'angle petita, ja que al ser una canonada circular i fent la hipòtesi anterior de simetria cilíndrica no és necessari dibuixar la canonada completa a més no facilitaria el temps de computació.

Per tant, la malla a generar ha de partir del següent esquema que serà, la geometria de la malla. La canonada tindrà un diàmetre de 8mm i la malla es representa el radi de 4 mm.

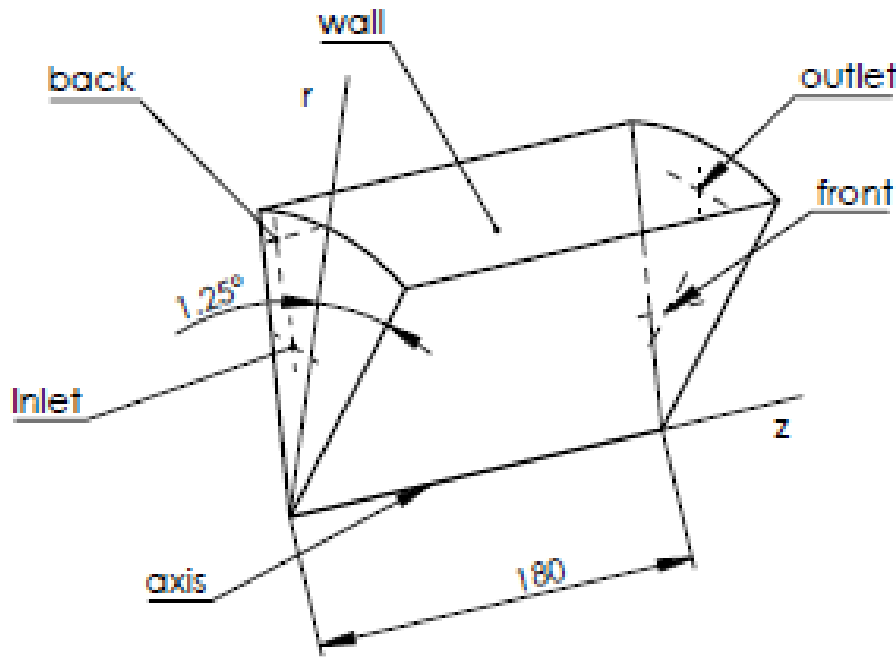


Figura 10.3: Resperesntació de la forma de la malla indicant les seves regions

A partir d'aquí es generarà la malla. Es recorda que la malla serà generada en coordenades cartesianes. Ja que els vèrtexs es generen en coordenades cartesianes. Utilitzant el $radi \cdot \sin(\theta)$ per l'eix de les z i $radi \cdot \cos(\theta)$ per l'eix de les y i la longitud de la canonada en l'eix de les x. El codi es pot localitzar a l'**annex 1.2.1.1** cas 1 del capítol 2.

En aquesta malla es pot observar un recurs que permet fer el programa que és la superposició de punts. Per generar la intersecció de punts o de dues arestes. A part es pot veure en el codi com generar un arc de radi concret. A continuació s'adjunta una taula per crear tota mena de formes corbes.

Taula 10.1: Tipus de formes dintre del diccionari BlockMeshDict [4]

Paraula clau	Descripció	Entrades addicionals
arc	Arc circular	Un punt d'interpol·lació
spline	Spline (conjunt de funcions polinòmiques definides a trossos)	Llista de punt
polyLine	Conjunt de línies	Llista de punts
BSpline	Spline, amb suport mínim respecte un grau, suavitat i partició de domini	Llista de punts
line	Linia recta	-

A continuació es declararan les regions tenint en compte la funció de la regió. Es recorda que sempre per declarar les regions s'ha de seguir la regla de la mà dreta on la normal de pla ha d'apuntar cap a fora.

10.1.2.1.2 transportProperties

Es generarà l'arxiu que fa referència al tipus de flux, serà un flux newtonià, tal com s'hi ha especificat a les hipòtesis. Amb una viscositat cinemàtica com s'ha especificat anteriorment de $\mu = 0.03 Pa \cdot s$. Es pot visualitzar a l'**annex 1.2.1.2**.

10.1.2.1.3 RASProperties

Aquest arxiu fa referència a les propietats físiques del problema on s'indicarà quin tipus de model fa referència al problema(**RASProperties**), si és turbulent o laminar, es recorda que si es volgués solucionar en mode turbulent, no seria correcte posar turbulent, ja que s'utilitza icoFoam que és per models laminars. Per tant s'ha d'especificar que no hi ha turbulències i que no imprimeixi els coeficients de turbulència indicant "off" ambdós llocs. Es pot consultar el codi a l'**annex 1.2.1.3**.

10.1.2.2 Carpeta 0

Dintre d'aquesta carpeta hi apareixeran. Les condicions inicials que fan referència a la pressió i a la velocitat.

10.1.2.2.1 U

Fa referència a la velocitat. Es calcularà la velocitat d'entrada a partir del caudal d'entrada (regió inlet).

$$V = \frac{Q}{\pi \cdot (D/2)^2} = \frac{25 \times 10^{-6}}{\pi \cdot (4 \times 10^{-3})^2} = 0.49736 m/s \quad (10.9)$$

En el codi apareix una nova condició de contorn **inletOutlet**. Aquesta funció és utilitzada quan la velocitat d'entrada és constant però es desconeix la velocitat que tindrà a mesura que passa per al conducte, però al saber la pressió de sortida és capaç de canviar el valor fixe de la velocitat d'entrada per la funció per **zeroGradient** que el que farà és adaptar-se a les condicions donades de la sortida, en aquest cas a 9000Pa, ja que la distribució canviarà de constant a parabòlica com es pot veure a la següent imatge.

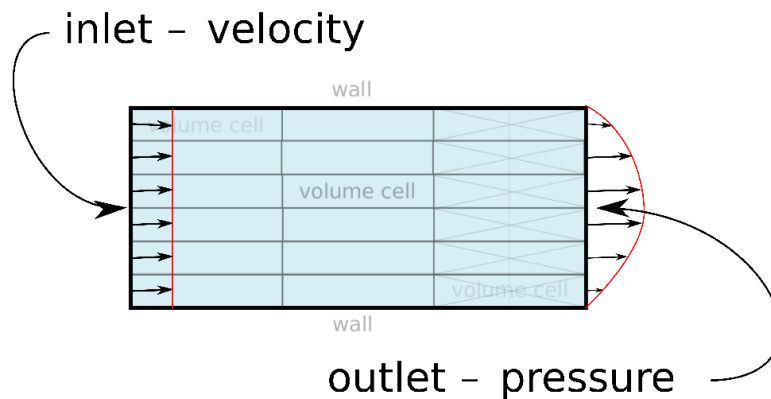


Figura 10.4: Esquema de funcionament de la funció inletOutlet [9]

El codi es pot trobar a l'**annex 1.2.1.4.1**.

10.1.2.2.2 p

Aquest codi que fa referència a la pressió no es sap la pressió a la zona d'entrada **inlet** però si es sap a la sortida que són uns 9000 Pa. El codi es pot trobar a l'**annex 1.2.1.4.2**.

10.1.2.3 System

Una vegada, determinades totes les condicions de contorn i haver definit el problema. Queda controlar els paràmetres de la solució, per això es generen els diferents arxius de la

carpeta **System**.

10.1.2.3.1 controlDict

El solucionador utilitzat és icoFoam. Se simularà començant del temps 0 al temps 0.2 s i és important controlar el nombre de courant, que mai pot ser superior a 1 tal com es va explicar al capítol 1 en aquest cas serà recomanable utilitzar un **deltaT**= 0.00005. Si es desitgen menys carpetes a generar es pot augmentar el factor de **writeInterval** en aquest cas es proposa 20. Es pot trobar a l'**annex 1.2.1.5.1**.

10.1.2.3.2 fvSchemes i fvSolution

Es proposen aquets dos arxius als annexes **1.2.1.5.2** i **1.2.1.5.3**.

10.1.3 Postproces

Una vegada creada la malla amb la funció blockMesh i resolt amb el solver icoFoam. Es pot passar al post-porcesing. Es recorda que s'ha de generar l'arxiu pipe.foam amb l'ordre **paraFoam -touch -builtin**. La malla generada ha de ser similar en aquesta:

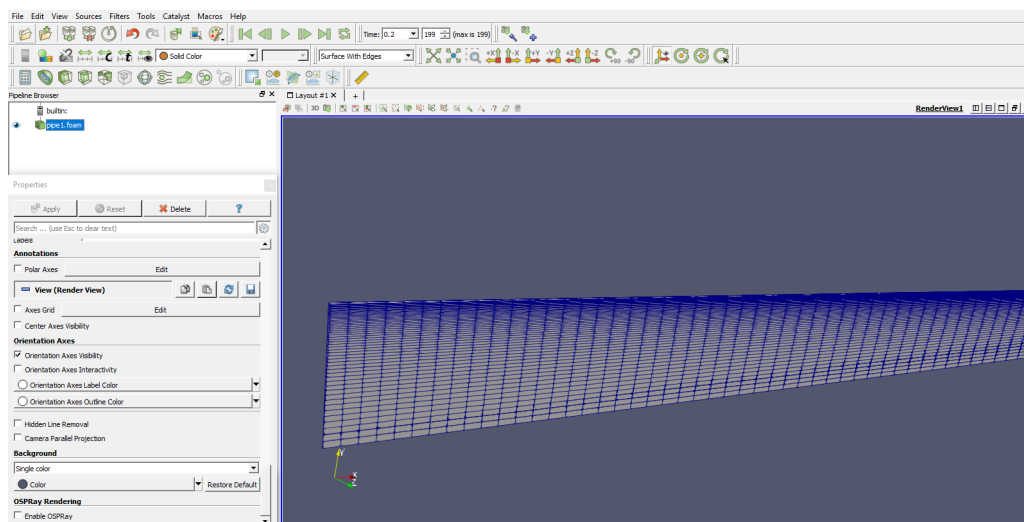


Figura 10.5: Malla generada

Una vegada obert l'arxiu **pipe.foam** des del paraview. Serà interessant veure el perfil generat, el perfil de velocitat una vegada s'hi ha desenvolupat correctament, ja que passa

de ser constant ha de tenir un flux parabòlic. Aquests seran els resultats del camp de velocitats i el camp escalar de pressió (m^2/s^2).

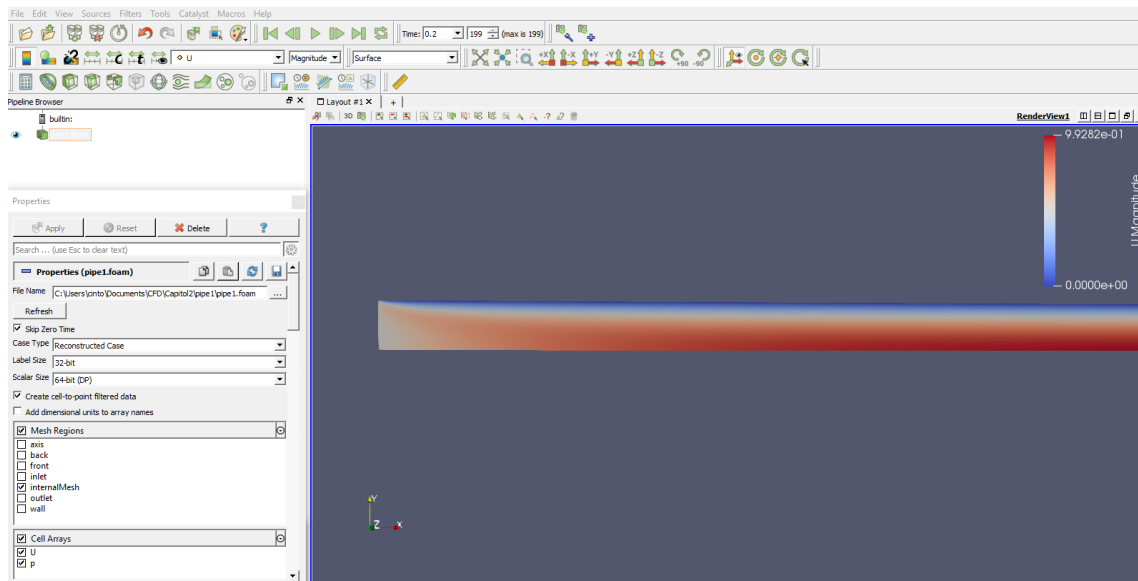


Figura 10.6: Camp de velocitats del cas

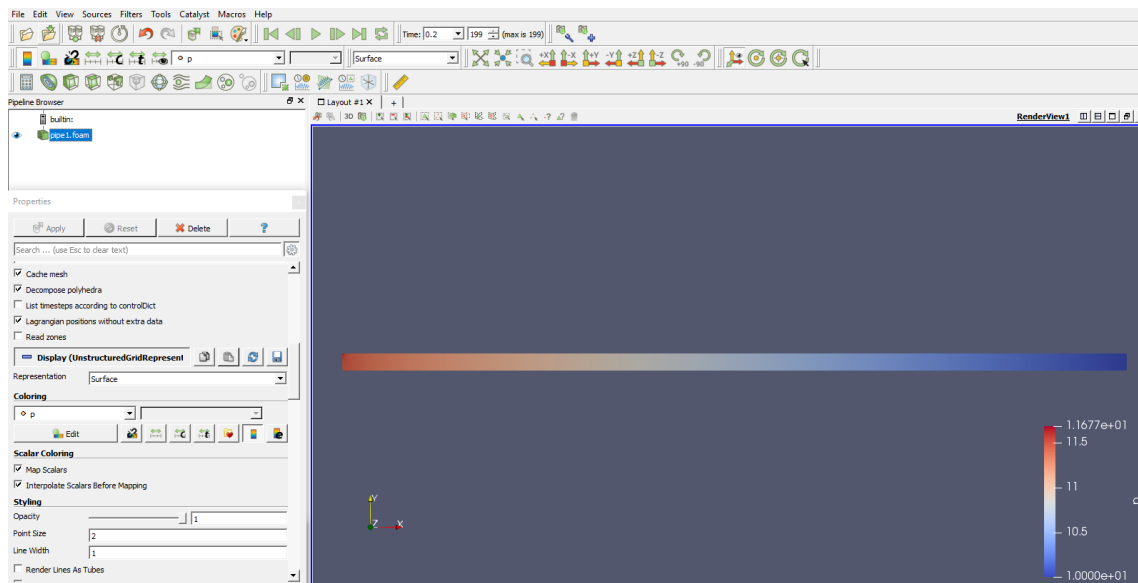


Figura 10.7: Camp escalar de pressió

A continuació apareixeran els perfils de pressions al llarg de la canonada. Es podrà observar al principi una petita corba, ja que és la fase de desenvolupament. Després ja segueix la linealitat esperada:

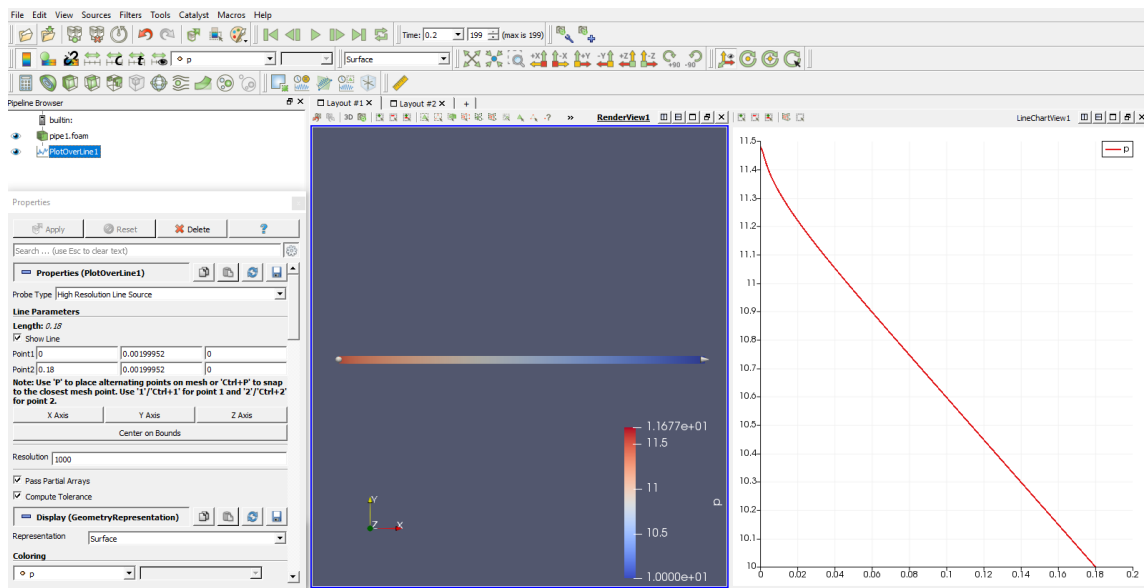


Figura 10.8: Variació de la pressió al llarg de la canonada

A la següent imatge apareixerà el perfil de velocitat a l'eix de les y (en coordenades cilíndriques r, en el cas analític). Una vegada desenvolupat a l'última secció en l'eix de les x (que en el cas analític seria eix de les z).

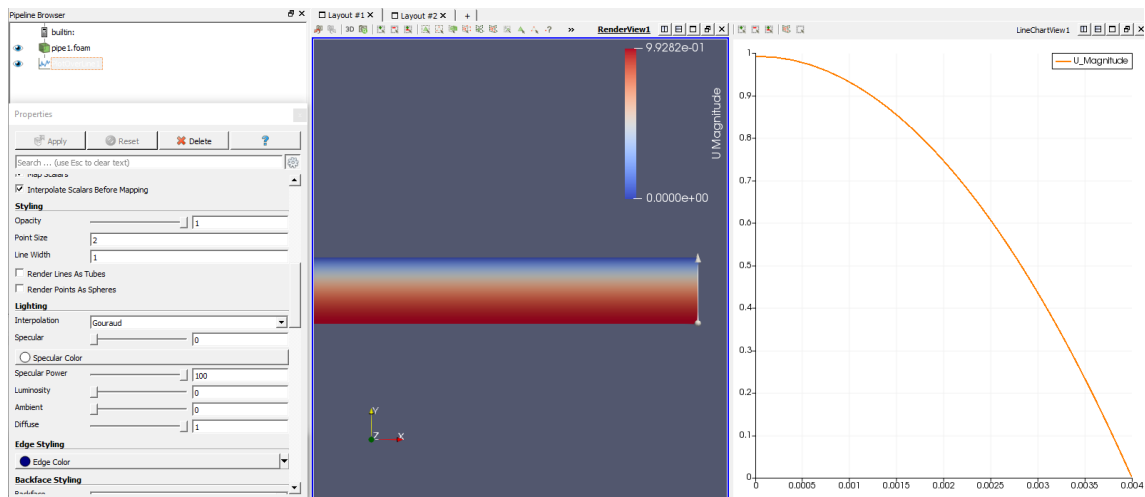


Figura 10.9: Variació de la pressió a l'inlet

Per comprovar la semblança amb el cas analític, primerament i de forma ràpida es pot observar que l'expressió analítica i la solució calculada per l'Openfoam totes dues segueixen una tendència parabòlica. A continuació, s'extreu l'expressió de la velocitat màxima amb la intenció de comparar les dues velocitats màximes. Però es troba amb la dificultat que la pressió a la regió inlet no és constant, ja que el flux no està desenvolupat.

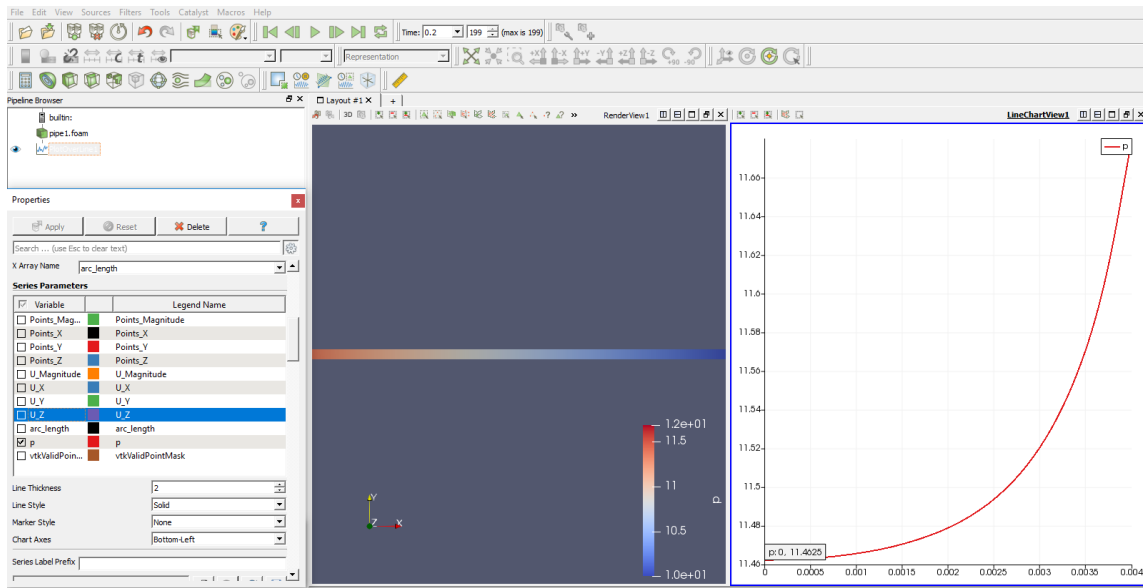


Figura 10.10: Variació de la pressió en l'eix de la y a la regió inlet.

Per tant s'agafa la pressió més pròxima en el centre de la canonada que seria el punt de la velocitat màxima. Es recorda que el valor de la imatge en el punt 0 s'ha de multiplicar per la densitat ($\rho = 900 \text{ kg/m}^3$).

$$v_{max} = \frac{\Delta p}{\Delta z} \frac{1}{4\mu} R^2 = \left(\frac{P_1 - P_2}{L} \frac{1}{4\mu} R^2 \right) = \frac{(11.4625 \cdot 900) - 9000}{0.18} \frac{1}{4 \cdot 0.03} \cdot (0.004)^2 = 0.975 \text{ m/s} \quad (10.10)$$

El programa té com a resultat 0.992 si es calcula l'error, és de l'ordre 1,74%

10.1.3.1 Càlcul del cabal de sortida

Un altre de forma de saber si no hi ha cap error, és calcular el cabal. Aquest cas s'ha tractat com si el fluid fos incompressible per tant el cabal d'entrada ha de ser el mateix que el de sortida (equació de la continuïtat). A continuació s'escriurà al terminal la següent funció:

```
postProcess -func "flowRatePatch(name=outlet)"
```

1

I al clicar “enter” començarà a calcular el cabal a la regió **outlet**.

$$Q_{inlet} = Q_{outlet} \quad (10.11)$$


```

surfaceScalarFields: phi
Executing functionObjects
surfaceFieldValue flowRatePatch(name=outlet) write:
    sum(outlet) of phi = 1.73556e-07

Time = 0.199

Reading fields:
    surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=outlet) write:
    sum(outlet) of phi = 1.73556e-07

Time = 0.2

Reading fields:
    surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=outlet) write:
    sum(outlet) of phi = 1.73556e-07

End

```

Figura 10.11: Imatge de linux amb el valor del cabal

Es pot observar que el cabal és $1.7355 \cdot 10^{-7} \text{ m}^3/\text{s}$, però aquest valor és per la secció d'angle per tant al tenir una secció d'angle $\theta = 2.5^\circ$, en aquest cas no serveix, ja que es desitja pels 360° al ser proporcional a causa de la hipotesis aplicada durant el problema de simetria cilíndrica, es dedueix que per 360° el valor és $2.49 \cdot 10^{-6} \text{ m}^3/\text{s}$ que si es converteix $24,9912 \text{ cm}^3/\text{s}$ amb un error de 0.0352%, comparant amb la dada del problema de $25 \text{ cm}^3/\text{s}$.

10.1.3.2 Localització de valors

De tant en tant, és necessari trobar un valor o un rang de valors, on estan situats a la malla. El programa ParaView té aquesta opció on indica quines cel·les o punts es situen aquests valors. Aquí es farà un exemple amb la recerca de pressions que estan entre $11.5 \text{ m}^2/\text{s}^2$ i $12 \text{ m}^2/\text{s}^2$. S'adjunta una imatge del procediment:

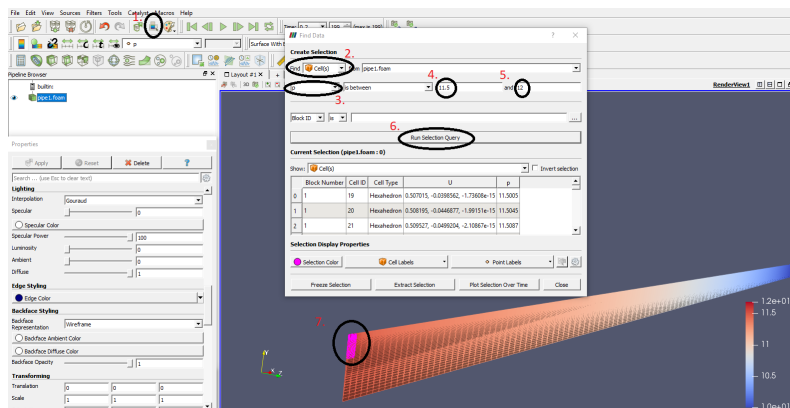


Figura 10.12: Selecció dels valors de les cel·les

1. Clicar la icona del punt 1 de la figura 10.12 i apareixerà una finestra secundària.
2. Elecció entre cel·les o punts.
3. Magnitud a cercar en aquest cas p (pressió).
4. En aquest cas s'ha triat un rang però podria ser un valor en concret o fins i tot màxim i mínim. A la casella 4 de la imatge valor mínim.
5. Valor màxim.
6. Clicar “Run Selection Query”.

A continuació apareixeran en rosa, les caselles que compleixen les condicions anteriors.

10.1.3.3 Graficar els residus de la simulació

La convergència de la solució a les simulacions per mètodes numèrics és un apartat molt important. Els valors van canviant a mesura que es genera cada iteració. Això vol dir que cada solució numèrica conté errors.

Si la solució no convergeix, normalment s'obtenen resultats no físics i una forma molt clara de mesurar la convergència és mesurar els residus, que representa l'error en valor absolut. Comparar el valor de la iteració $n+1$ amb la iteració n i veure com aquest valor va disminuint cap a un valor cada cop més petit (el residu tendeix i s'aproxima més a 0 en cada iteració), aquest és l'argument per detectar que la solució va convergint.

Quan els residus aconseguixen valors dintre de les toleràncies definides a l'arxiu **fvso-lution** en concret aquesta és la línia de codi que s'aplica les toleràncies, més concretament la de la pressió. També es poden especificar d'una altra forma, com es podrà veure als pròxims capítols: [4]

```

p                                     1
{                                     2
  solver                             PCG;                               3
  preconditioner                      DIC;                               4
  tolerance                           1e-07;                           5
  relTol                              0;                                6
}                                     7

```

En aquest apartat s'ensenyarà a generar a representar els valors residuals en funció del temps. Aquest gràfic podrà ajudar a comprendre a l'usuari si la solució convergeix i en quina rapidesa ho fa. Cal destacar que una solució que és convergent, no garanteix uns bons resultats. Els factors com un mal plantejament del problema, una malla mal generada o fins i tot una malla gruixuda, pot generar errors en els resultats.

S'ha d'incloure dintre de les funcions de l'arxiu **controlDict**, el següent codi: [4]

```
functions                                     1
{                                             2
#includeFunc residuals                        3
}                                             4
```

A continuació aplicar la següent ordre:

```
rm -rf postProcessing                        1
icoFoam > log &                             2
```

Per visualitzar els resultats com s'ha explicat anteriorment s'ha de fer doble clic al programa de l'escriptori descarregat a la guia **Xming** i a partir d'aquí introduir la següent funció a la imatge de Linux. La **-l** del següent codi és opcional, és per generar amb escala logarítmica a l'eix de les y.

```
foamMonitor -l postProcessing/residuals/0/residual.dat 1
```

A continuació de forma automàtica és generarà un gràfic amb els resultats, com els següents:

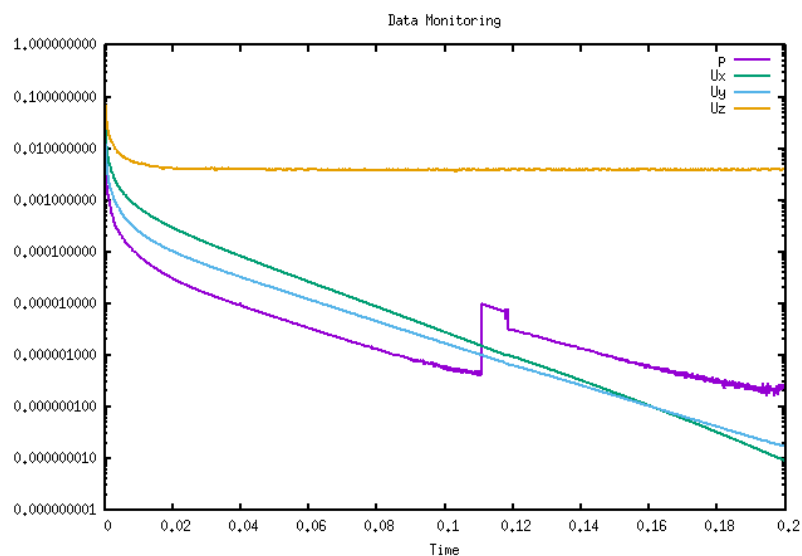


Figura 10.13: Gràfica del residu en funció del temps.

10.2. Cas 2: Flux en una canonada anular

Al cas següent, es posarà com a exemple una canonada anular amb una diferència de pressions coneguda. $P_1=50663$ Pa com a pressió a l'entrada i $P_2=10000$ Pa com a pressió de sortida, $\rho = 1000 \text{ kg/m}^3$, $\mu = 0.032 \text{ Pa} \cdot \text{s}$. S'estudiarà el cas a partir d'una secció d'aquesta. Amb la geometria següent:

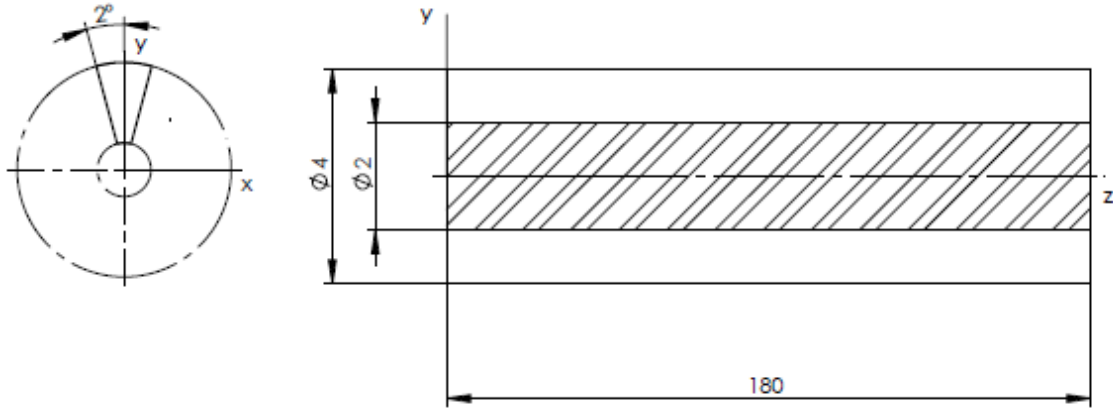


Figura 10.14: Esquema del cas 2.2

Es farà referència $L=180$ com la longitud del conducte, R_i fa referència al radi interior, R_0 fa referència al radi de la canonada, $\frac{\Delta P}{\Delta x}$ fa referència a la diferència de pressions.

10.3. Resolució analítica

Utilitzant les mateixes hipòtesis que el cas anterior. Queda la mateixa expressió de distribució de velocitats. [3]

$$V_z = \frac{\Delta P}{\Delta x} \frac{r^2}{4\nu} + C_0 \ln r + C_1 \quad (10.12)$$

Però les condicions de control varien. Les noves condicions de contorn són aquestes: [3]

$$r = R_0; V_x = 0 \quad (10.13)$$

$$r = R_i; V_x = 0 \quad (10.14)$$

$$\frac{\Delta P}{\Delta x} = 0 \quad (10.15)$$

Aplicant les equacions de contorn s'obté:

$$0 = \frac{\Delta P}{\Delta x} \frac{R_0^2}{4\mu} + C_0 \ln R_i + C_1 \quad (10.16)$$

$$0 = \frac{\Delta P}{\Delta x} \frac{R_i^2}{4\mu} + C_0 \ln R_i + C_1 \quad (10.17)$$

Si es resten les dues equacions anteriors. S'extreu que:

$$0 = \frac{\Delta P}{\Delta x} \frac{1}{4\mu} (R_0^2 - R_i^2) + C_0 (\ln \frac{R_0}{R_i}) \quad (10.18)$$

$$C_0 = \frac{\Delta P \cdot (R_0^2 - R_i^2)}{\Delta x \cdot 4\mu \cdot \ln \frac{R_0}{R_i}} \quad (10.19)$$

I substituint a 1:

$$C_1 = -\frac{\Delta P}{\Delta x} \cdot \frac{R_0^2}{4\mu} + \frac{\Delta P}{\Delta x} \cdot \frac{R_0^2 - R_i^2}{4\mu} \cdot \frac{\ln R_0}{\ln(\frac{R_0}{R_i})} \quad (10.20)$$

Per tant la funció de la velocitat en z.

$$V_z = \frac{\Delta P}{\Delta x} \frac{1}{4\mu} \left[(r^2 - R_0^2) + (R_0^2 - R_i^2) \frac{\ln \frac{R_0}{r}}{\ln \frac{R_0}{R_i}} \right] \quad (10.21)$$

La velocitat màxima entre dos cilindres s'obté aplicant $\frac{dV_x}{dr} = 0$ on i treure factor comú:

$$\frac{\Delta P}{\Delta x} \frac{1}{4\mu} \left[2r - \frac{(R_0^2 - R_i^2)}{\ln(\frac{R_0}{R_i})} \frac{1}{r} \right] = 0 \quad (10.22)$$

On s'extreu que el radi amb la velocitat màxima serà:

$$r_{V_{zmax}} = \left(\frac{(R_0^2 - R_i^2) \frac{1}{2}}{\ln\left(\frac{R_0}{R_i}\right)} \right)^{1/2} \quad (10.23)$$

10.3.1 Preprocés amb OpenFOAM

Abans de res cal recordar que es seguirà la mateixa estructura de carpetes que el cas anterior. Per tant es pot simplificar, copiant la carpeta i canviant el nom. Així únicament caldrà adaptar el cas canviant codi, des del programa gedit, descarregat en capítols anteriors.

10.3.1.1 Generació de la malla

La malla a generar serà una porció de 2º de la circumferència que és la canonada, però retallant el que seria el cilindre d'enmig. Amb una longitud de 180mm. En aquest apartat es veurà com es generen diferents arcs i sobretot la importància que té l'ordre dels vertexes a l'hora de generar les diferents regions. Es recorda que aquest arxiu va dintre de la carpeta **constant** i dintre de **polyMesh**.

Amb l'esquema anterior s'han determinat els punts i amb l'ordre correcte. Per generar l'hexaedre que tindrà aquesta forma. Seguint l'ordre determinat, ja que pot portar a errors a l'hora de generar la malla.

```
blocks 1
( 2
hex (3 2 1 0 7 6 5 4) (20 1 50) simpleGrading (0.1 1 10) 3
); 4
```

Una vegada generat l'hexaedre, es generen els arcs. Es recorda que s'ha de posar un tercer punt per poder generar l'arc desitjat per poder interpolar.

A continuació es descriuran les següents regions. Com s'ha fet anteriorment és recomanable seguir la llei de la mà dreta per generar les regions. Que apareixen a la imatge anterior. Es pot consultar el codi a l'**annex 1.2.2.3.1**.

10.3.1.2 Condicions inicials

Dintre de la carpeta **0**, s'introduiran els valors inicials, com bé s'ha explicat. Per tant s'indicaran els valors de pressió i velocitat dintre de la carpeta corresponent.

10.3.1.2.1 U

Fent referència a la velocitat. Es pot indicar a cada regió com s'ha de comportar. Al produir-se el moviment a causa de la diferència de pressions no tenim un valor per la velocitat, per tant serà calculat pel programa (fa referència a les regions on hi apareix la funció **zeroGradient**), però si es sap que la velocitat a les parets de la tuberia serà 0 m/s, ja que no hi ha reliscament amb el fluid i les parets(funció **fixedValue**) i finalment la funció **wedge** fa referència al fet que no hi ha unes parets físiques, però que limiten la malla (hipòtesis de simetria cilíndrica). El plantejament de la velocitat en aquest cas es pot trobar a l'**annex 1.2.2.1.1**.

10.3.1.2.2 p

Fa referència a la pressió. Utilitzant les mateixes funcions, sabent que la pressió a la regió inlet $P1=50.663 \text{ m}^2/\text{s}^2$ i a la regió outlet $P2= 10 \text{ m}^2/\text{s}^2$. La pressió a les parets de les canonades seran calculades pel programa i les regions, front i back la funció **wedge**. Es pot consultar a l'**annex 1.2.2.1.2**.

10.3.1.2.3 RASProperties i transportProperties

S'utilitzarà els mateixos arxius. Cal recordar que:

- **RASProperties**, s'utilitzarà un model laminar i la mostra de coeficients i la turbulència s'han d'indicar com a "off". Es troba a l'**annex 1.2.2.2.1**.
- **transportProperties**, S'utilitzarà un model newtonià de transport del fluid i una viscositat cinemàtica, $\mu = 0.032 \text{ Pa} \cdot \text{s}$ es recorda que no es pot posar unitats de pressió, com s'ha explicat anteriorment i simplement s'ha de dividir per la densitat ($\rho = 1000 \text{ m}^3/\text{s}$). Es troba a l'**annex 1.2.2.2.2**.

10.3.1.3 System

Seguint l'esquema indicat al cas 1 dintre de la carpeta constant. Hi ha tres arxius:

- **controlDict**, s'utilitzarà el "solver" icoFoam. La simulació començarà al 0 s i acabarà al 0.2 s, deltaT serà 0.00005 al ser tan petita s'incrementarà el writeinterval per aconseguir que les carpetes generades per al programa no siguin tantes i apareguin a cada centèsima (0.00005 x 200=0.01). A l'**annex 1.2.2.3.2** es pot torbar el codi sencer.
- **fvSchemes** i **fvSolution**, seran els mateixos procediments de solucionar que el cas anterior. En els **annexos 1.2.2.3.3 i 1.2.2.3.4** es pot trobar el codi respectivament.

10.3.1.4 Postprocés

Una vegada s'han modificat tots els arxius per adaptar-los al cas. És convenient generar la malla com s'ha fet anteriorment amb la funció **blockMesh** i a continuació utilitzar la funció **icoFoam**. Una vegada compilat el programa crear un arxiu temporal amb la funció **paraFoam -touch -builtin**. Obrir amb el programa paraview i cercar l'arxiu "annularpipe.foam".

A continuació es calcularà de forma analítica la velocitat màxima: Si es substitueix a l'equació anterior:

$$r_{V_{z_{max}}} = \left(\frac{(0.002^2 - 0.004^2) \frac{1}{2}}{\ln\left(\frac{0.004}{0.002}\right)} \right)^{1/2} = 0.002942137m = 2.9442137mm \quad (10.24)$$

Si es substitueix a l'equació de la V_z :

$$V_z = \frac{(10000 - 50663)}{0.18} \frac{1}{4 \cdot 0.032} \left[(0.0029421^2 - 0.004^2) + (0.004^2 - 0.002^2) \frac{\ln \frac{0.004}{0.0029421}}{\ln \frac{0.004}{0.002}} \right] = 3.5760m/s \quad (10.25)$$

Si es compara amb els resultats calculats amb l'Openfoam:

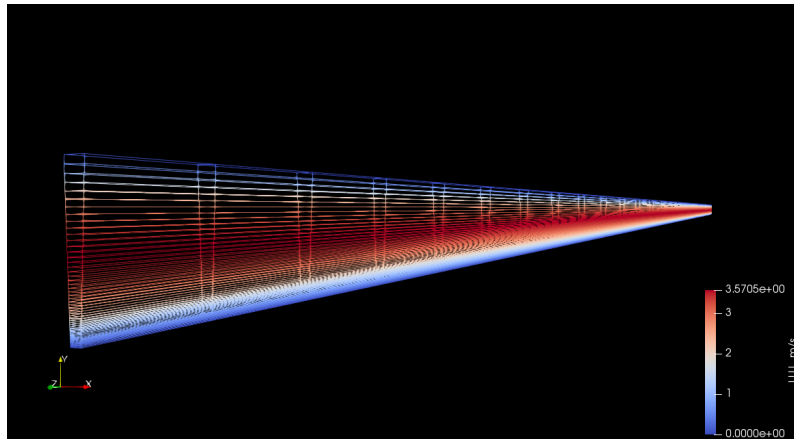


Figura 10.15: Resultats de velocitat

Es pot observar que el valor màxim de la velocitat en aquest cas és de 3.5705 m/s. Per tant té un 0.15% d'error. Ara es comprovarà que tingui la velocitat en V_z segueixi la mateixa funció que la trobada:

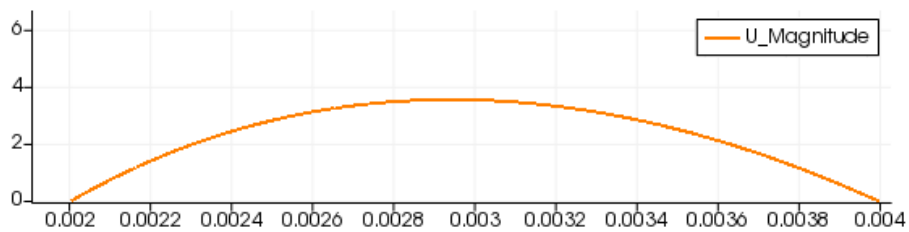


Figura 10.16: Gràfica de la variació de la velocitat en funció de l'eix de les y

Com es pot veure la velocitat màxima no coincideix amb el valor 0.003m com s'ha calculat analíticament, que està una mica descentrat $r_{V_{max}} = 0.0029421$ m.

Utilitzant la visualització dels vectors amb la funció **glyph**, ja explicada.

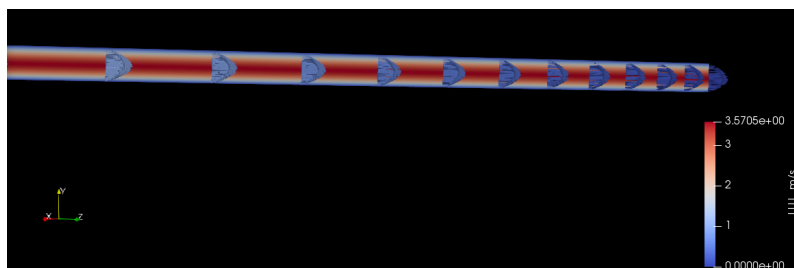


Figura 10.17: Visualització dels vectors

En el cas de les pressions, es comprova que efectivament té una tendència lineal decreixent. Tal com s'ha demostrat a la resolució analítica.

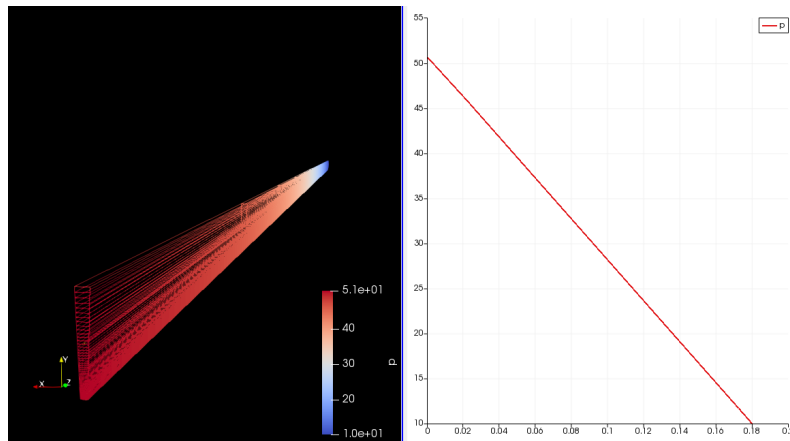


Figura 10.18: Gràfica de la variació de la pressió en funció de l'eix z

10.3.1.5 Càlcul del cabal

En aquest apartat es comprovarà el cabal calculat per Openfoam i el cabal analític. L'expressió que determina el cabal és la següent:

$$Q = \frac{\pi}{8\mu} \left[-\frac{dp}{dx} \right] \cdot \left[R_0^4 - R_i^4 + \frac{(R_0^2 - R_i^2)^2}{\ln\left(\frac{R_i}{R_0}\right)} \right] \quad (10.26)$$

Si se substitueixen a les dades del problema:

$$Q = \frac{\pi}{(8 \cdot 0.32)} \cdot \left(-\frac{(10000 - 50663)}{0.18} \right) \cdot \left(0.004^4 - 0.002^4 + \frac{(0.004^2 - 0.002^2)^2}{\ln\left(\frac{0.002}{0.004}\right)} \right) = 8.94 \cdot 10^{-5} m^3/s \quad (10.27)$$

Cal dir que aquest resultat, és el cabal de tota la geometria i la geometria dissenyada amb Openfoam no és la mateixa per això es calcularà el cabal per unitat d'àrea. Per tant:

$$\frac{(8.94 \cdot 10^{-5})}{\pi \cdot (0.004^2 - 0.002^2)} = 2.371408652 m/s \quad (10.28)$$

Per tal de calcular el cabal amb el programa s'utilitza la següent funció, a la imatge de Linux en el directori del cas:

```
postProcess -func "flowRatePatch(name=inlet)"
```

1

Una vegada ha acabat el resultat és extret de l'Openfoam és $-9.932 \cdot 10^{-07} \text{ m}^3/\text{s}$.

```
Time = 0.2

Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=inlet) write:
  sum(inlet) of phi = -9.932e-07

End
```

Figura 10.19: Resultat del cabal en m^3/s

Primerament cal dir, que el cabal és negatiu però realment és un resultat correcte, ja que el cabal té un sentit contrari al vector normal de la regió **inlet**, en canvi a la regió **outlet**, si es volgués calcular, el signe apareixerà positiu. Es pot comprovar únicament canviant de la línia anterior la paraula inlet per outlet i sortirà el mateix resultat canviat de signe.

Aquest cabal seria per una secció circular de 4° . Per tant es calcularà el cabal per unitat d'àrea per tal de tenir-ho igual que l'equació 10.28.

$$\frac{9.932 \cdot 10^{-7}}{\frac{1}{2} \cdot (4 \cdot \frac{\pi}{180}) \cdot (0.004^2 - 0.002^2)} = 2.3710903 \text{ m/s} \quad (10.29)$$

Si es calcula l'error, on el resultat analític és 10.28 i l'extreta de l'Openfoam 10.29 generen un error de 0.013%.

10.4. Cas 3 : Secció anular amb el cilindre interior amb velocitat angular

En aquest cas la geometria serà canviada, el cilindre anterior rotarà en l'eix de les x, mentre que la canonada serà fixa. A continuació s'adjunta un dibuix per entendre millor la geometria.

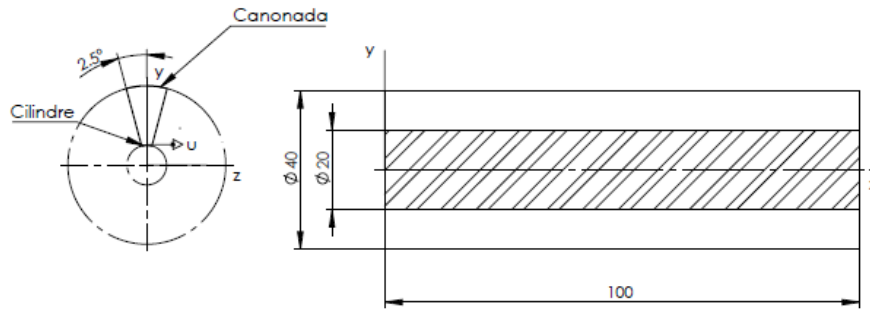


Figura 10.20: Esquema del problema resolt, les unitats longitudinals en mm.

10.4.1 Resolució analítica

10.4.1.1 Hipòtesis

Les hipòtesis al problema seran les següents:

- $\rho = cte$, per tant un fluid incompressible.
- $\frac{\partial \vec{u}}{\partial t} = 0$, per tant serà un règim permanent.
- $V_\theta = f(r)$, flux unidimensional. On la velocitat del fluid dependrà únicament del radi.
- $v_r = v_x = 0$, flux unidireccional bidimensional.

Per tant es considera que els cilindres són infinitament llarg i de règim permanent, la velocitat varia únicament en direcció radial.

10.4.1.2 Condicions de contorn

$$r = R_i; \quad V_{\theta i} = \omega_i R_i \quad (10.30)$$

$$r = R_0; \quad V_{\theta 0} = 0 \quad (10.31)$$

10.4.1.3 Resolució

Partint de les equacions de continuïtat i de Navier-Stokes, en coordenades cilíndriques. S'obté que [3]:

- Equació de la continuïtat:

$$\frac{1}{r} \frac{\partial}{\partial \theta} = 0 \rightarrow \rho \frac{\partial V_\theta}{\partial \theta} = 0 \quad (10.32)$$

- De l'equació de Navier-Stokes:

En direcció radial:

$$\rho \frac{V_\theta}{r} = \frac{\partial P}{\partial r} \quad (10.33)$$

En direcció axial:

$$0 = \frac{\partial P}{\partial z} \quad (10.34)$$

En direcció angular:

$$0 = \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} (r V_\theta) \right) \quad (10.35)$$

$$\frac{1}{r} \frac{d}{dr} (r V_\theta) = C_1$$

$$\frac{d}{dr} (r V_\theta) = C_1 r$$

$$r V_\theta = C_1 \frac{r^2}{2} + C_2$$

$$V_\theta = \frac{C_1}{2} r + \frac{C_2}{r} \quad (10.36)$$

Aplicant les condicions de contorn expressades anteriorment 10.30 i 10.31, s'obté C_1 i C_2 i substituint s'obté l'equació de com varia la velocitat angular en funció del radi. [3] [2]

$$V_\theta = \frac{R_i^2 \omega_i}{R_0^2 - R_i^2} \left(\frac{R_0^2}{r} - r \right) \quad (10.37)$$

10.4.2 Preprocés amb OpenFoam

A continuació apareixerà un esquema a seguir en aquest cas.

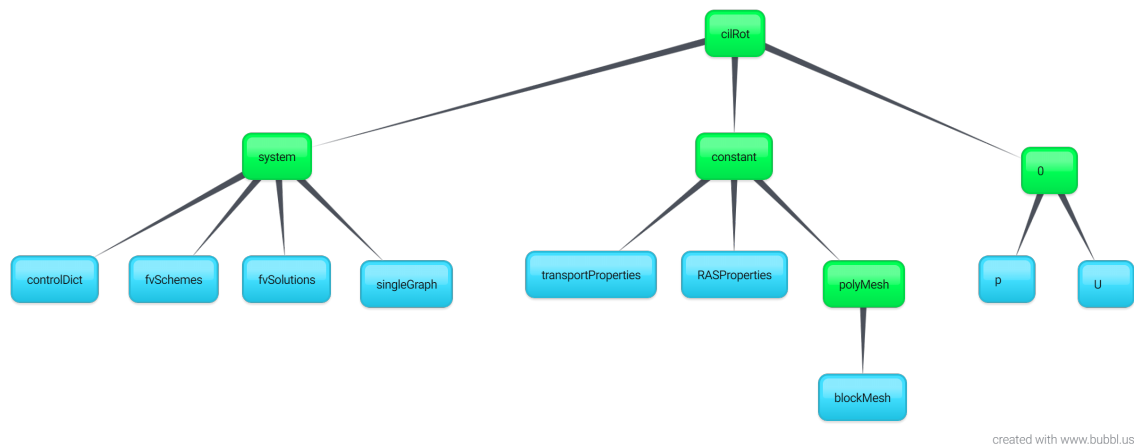


Figura 10.21: Esquema de fitxers del cas 3. En verd apareixen les carpetes mentre que en blau són els fitxers

10.4.2.1 Carpeta constant

10.4.2.1.1 Creació de la malla

Com sempre es generarà un arxiu amb el nom de **blockMeshDict** dintre de la carpeta **constant** i dintre de **polyMesh**. També es pot ficar dintre de la carpeta **system** si es desitja, on es crearà la malla. Com que no varia la velocitat V_θ en funció de l'angle, s'agafarà una fracció de la circumferència. Es seguirà al mateix esquema que la malla del cas 2. Cal recordar que aquest cop en agafar una fracció tan petita de la circumferència, s'eliminaran els arcs i seran substituïts per línies rectes per tal d'optimitzar millor la malla. A continuació s'adjunta una imatge on quedaran definides totes les regions i a més quin ordre seguiran per indicar els punts.

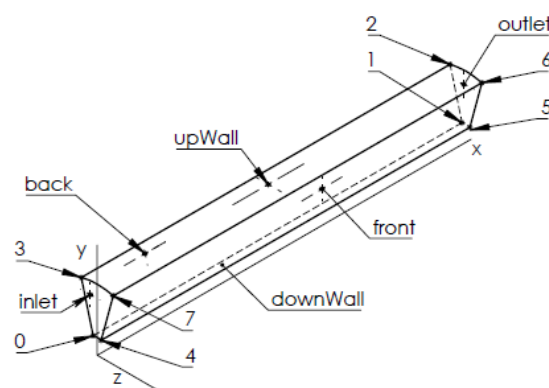


Figura 10.22: Esquema de la malla a crear, on apareixen el nom dels punts i de les regions.

A part cal repetir que l'angle no pot ser superior a 5° , ja que si no el programa dóna error, ja que, s'utilitza la funció **wedge**. Utilitzada per casos on la malla continuaria amb

una estructura periòdica a la generada. A més optimitza el nombre de cel·les de la malla. S'utilitza en casos de 2D. Les regions descrites per aquesta funció són les regions, front i back, on el tub continuaria fins a generar una circumferència completa. Les regions, outlet i inlet, són regions declarades com a **patch**, que són regions on es poden introduir dades del problema però no són un mur de forma física. Finalment es declara les regions upWall i downWall com a **wall**, ja que representaran murs físics. Es pot consultar el codi a l'**annex 1.2.3.1.1**.

10.4.2.1.2 RASProperties i transportProperties

- **RASProperties**, s'utilitzarà el RASModel laminar i els paràmetres de turbulència en mode off i el mateix per als coeficients. El codi es pot consultar a l'**annex 1.2.3.1.2**.
- **transportProperties**, on es declararà un model de transport newtonià i el valor de la viscositat cinemàtica, dintre del fitxer apareix amb el nom de nu. El codi apareix a l'**annex 1.2.3.1.3**.

10.4.2.2 Carpeta 0

10.4.2.2.1 U

Fa referència a la velocitat, es declararà de la següent manera.

- A regió **downWall**, es declararà una velocitat en l'eix de les z de 0.1 m/s. Utilitzant la funció fixedValue. Es pot comprovar que genera el mateix resultat utilitzant una funció més obvia que seria **rotatingWallVelocity** que seria declarar-ho de la següent manera.

```

downWall                                     1
{                                             2
  type           rotatingWallVelocity;      3
  origin         (0 0 0);                   4
  axis           (0 0 0.1);                 5
  omega          -10;                       6
}                                             7

```

origin fa referència al punt de referència de la rotació. Mentre que **axis** seria l'altre punt per generar l'eix de la rotació, clarament **omega** fa referència a la velocitat angular en rad/s, es negativa, ja que així girarà en sentit a favor de les agulles del rellotge.

- **fixedValue**, s'utilitzarà la regió **upWall** per determinar que el mur no es mourà.
- Les regions **inlet** i **outlet**, s'aplicarà la funció **zeroGradient**.

- La condició de contorn de les regions **front** i **back** se declararan com a **wedge**, utilitzada per a casos en 2D.

Es pot consultar a l'**annex 1.2.3.2.1**.

10.4.2.2.2 p

Fent referència a la pressió, es declararà una pressió inicial interna de tota la malla de $0 \text{ m}^2/\text{s}^2$. A les altres regions:

- A les regions **upWall** i **downWall** s'utilitzarà la funció **zeroGradient**, ja que no se saben els valors de les pressions en els dos dels casos.
- A les regions **inlet** i **outlet** s'utilitzarà la funció **fixedValue** amb un valor de $0 \text{ m}^2/\text{s}^2$, ja que el cas no està pensat perquè tingui una diferència de pressions en l'eix de les z.
- Finalment, a les regions **front** i **back** s'utilitza la funció **wedge**, pels mateixos motius que en el cas de velocitat.

Es troba a l'**annex 1.2.3.2.2**.

10.4.2.3 Carpeta system

- **ControlDict**, el problema serà solucionat pel solver **icoFoam**. Començant pel temps 0 i acabat al temps 0.2 s. Amb una **deltaT** de 0.0005 i cada 50 **deltaT** escriurà una carpeta dintre del cas. S'afegiran diferents funcions per tal de poder comparar el perfil de velocitats. Les funcions es declaren al final de l'arxiu de la següent forma:

```

functions                                     1
{                                              2
#includeFunc components(U)                    3
#includeFunc singleGraph                      4
#includeFunc streamlines                      5
}                                              6

```

► **components (U)**, dels resultats generats, en forma de vector aquesta funció crea un arxiu separant les components. En aquest cas les descompondrà amb U_x , U_y , U_z .

► **streamlines**, per poder visualitzar des del paraview.

► **singleGraph**, aquesta funció s'utilitza per a la generació de valors per cada component que hi ha dintre de totes les carpetes i poder graficar amb el programa

gnuplot. Els resultats i comparar-los per exemple amb una altra gràfica. On s'explicarà més en detall al post procés. En cas de dubte es pot consular el codi complet a l'annex 1.2.3.3.1.

- **fvSchemes** i **fvSolution**, són igual als altres casos. A l'annex 1.2.3.3.2 i 1.2.3.3.3 respectivament.
- **singleGraphs**, com s'ha comprovat aquest fitxer és nou, no ha aparegut a cap altre exercici i servirà per ajudar a comprovar la semblança del perfil de la velocitat en z del resultat generat amb OpenFoam i amb la solució analítica. Aquest arxiu es pot exportar del mateix OpenFoam. Per fer-ho cal primerament des de la imatge de Linux situar-se en el directori del cas amb les comandes generades. I introduir el següent:

```
cp -r $FOAM_ETC/caseDicts/postProcessing/graphs/singleGraphs system 1
```

S'utilitza la funció `cp -r` que serveix per copiar aquest fitxer a dintre de la carpeta `system`. Per defecte sempre tindrà tendència a representar la velocitat en x, cosa que en aquest cas no es d'interès. Per això s'utilitza la funció “components” (explicada anteriorment en el fitxer `controlDict`) perquè ens generi una carpeta per a cada component de la velocitat **deltaT**. Dintre del codi de l'arxiu **singleGraphs** adaptar-lo perquè llegeixi la carpeta **Uz**. També s'ha de definir igual que en `paraView`, la línia que vol que grafiqui amb les coordenades que apareixen “start” i “end”. A continuació s'adjunta el fragment amb les línies clau:

```
start (0.05 0.01 0); //Punt inicial 1
end (0.05 0.02 0); // Punt final. 2
fields (Uz p); //Carpetes a llegir. 3
```

Amb el punt inicial i el punt final es generarà una recta on es recolliran les dades, de les carpetes que s'han escollit. El codi es troba a l'annex 1.2.3.3.4.

10.4.2.4 Postprocés

Una vegada s'han generat tots els arxius es pot continuar, generant la malla, utilitzant la funció **blockMesh** i a procedir a utilitzar el “solver” `icoFoam`. Una vegada s'ha acabat la simulació. Cal generar el fitxer amb l'extensió `.foam`, amb la funció **paraView -touch -builtin**. Els resultats obtinguts són:

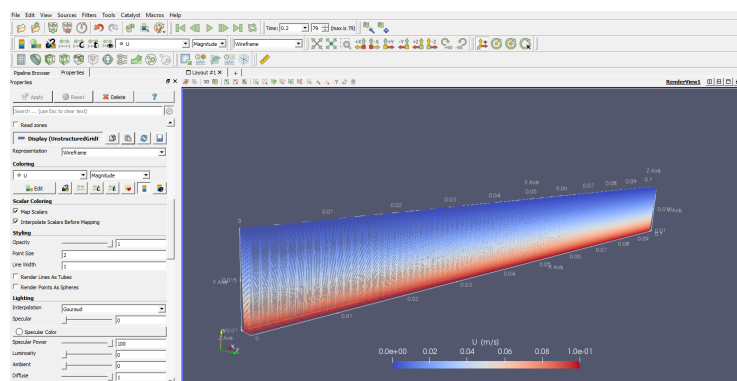


Figura 10.23: Solució del perfil de velocitats

Com es pot observar únicament existeix variació de la velocitat en l'eix de les Y. A continuació tal com s'ha explicat en els casos anteriors es crearà una gràfica per observar com varia velocitat en l'eix de les y.:

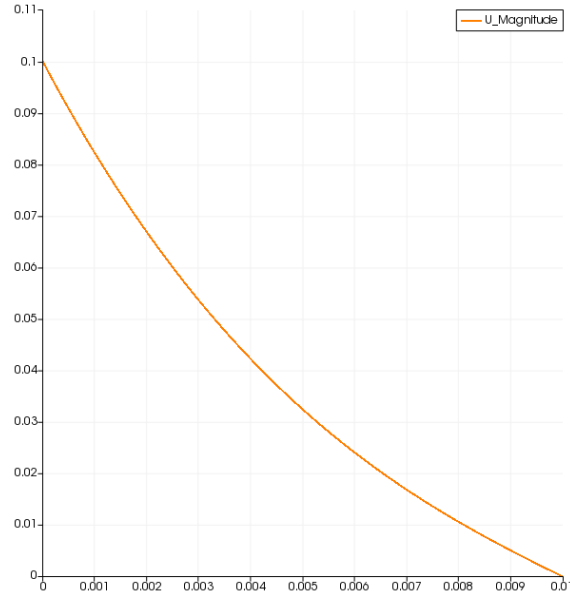


Figura 10.24: Variació de la velocitat en funció del y

S'ha d'entendre que la resolució analítica s'hi han utilitzat coordenades cilíndriques, i l'Openfoam utilitza coordenades cartesianes per tant els eixos mai seran els mateixos però la r en aquest cas equivaldria a la y. Tal com està disposada la malla. Si és substitueix a l'equació de la velocitat 10.37 amb els valors de $R_0 = 0.02$, $R_i = 0.01$ i $w = 10$:

$$V_{\theta} = \frac{0.00133333333333}{r} - 3.33333333 \cdot r \quad (10.38)$$

A continuació, a través de la imatge de Linux amb el directori dintre del cas. Amb els fitxers creats amb la funció singleGraph es procedirà a graficar el perfil de les velocitats generat per Openfoam i l'analític 10.37.

```
gnuplot 1
gnuplot> set style linespoints 2
gnuplot> plot "postProcessing/singleGraph/0.1975/line_Uz_p.xy" u 1:2 3
gnuplot> replot [0.01:0.02] 0.001333333333/x -3.333333333*x 4
gnuplot> set xlabel "Dist_[m]" 5
gnuplot> set ylabel "Velocitat_[m/s]" 6
```

La primera comanda és per entrar al gnuplot des del terminal d'Ubuntu i del directori del cas. La segona comanda serveix per unir els punts. La tercera indica el directori del qual es vol graficar la velocitat. Llavors s'enumeraran els eixos. La quarta fa referència a l'expressió 10.38 i el rang. Les modificacions a l'hora de generar la gràfica. La solució proposada és la següent:

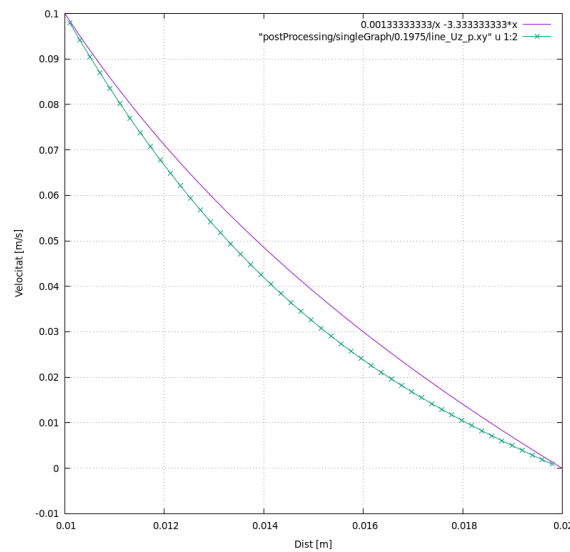


Figura 10.25

En lila és la gràfica de forma analítica mentre que en verd amb les marques correspon a la solució de l'Openfoam.

10.4.2.5 Càlcul dels esforços tangencials

Com bé es va explicar al capítol anterior amb el solver icoFoam, no es poden calcular els esforços tangencials. Per tant un “solver” que sí que és capaç de fer-ho és el simpleFoam per tant, s'ha de crear un model per executar-lo. Ja que s'ha creat un model amb simpleFoam es visualitzarà damunt la resolució analítica de la velocitat quina hi ha diferència.

A continuació s'explicarà una la diferència de codi en:

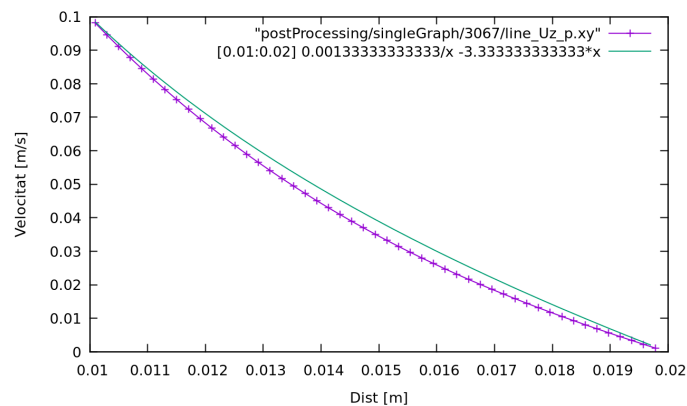


Figura 10.26: Comparació de la solució analítica i la solució del openfoam amb el solver simpleFoam

L'esquema de l'exemple seguirà el mateix esquema utilitzat al cas 2 del capítol 1. Com es pot veure s'obtenen resultats coherents, a més la solució convergeix abans dels cicles esperats. El codi proposat serà als **annexos 1.2.4** on també apareix l'esquema del cas. Els resultats són els següents:

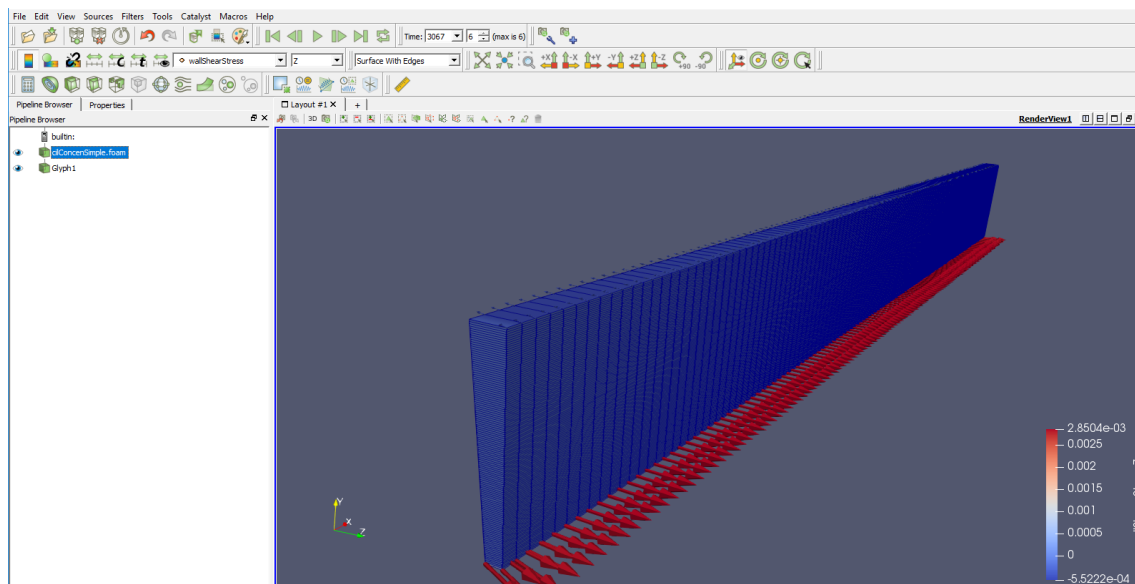


Figura 10.27: Visualització dels vectors

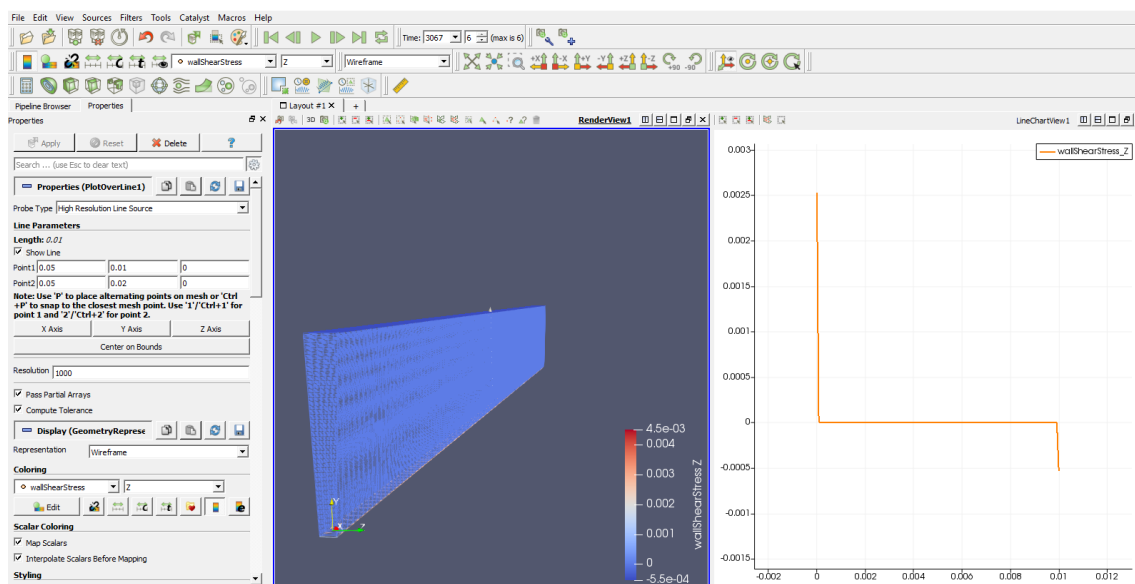


Figura 10.28: Gràfica dels esforços tangencials en l'eix z.

Es pot observar que els esforços tangencials a la regió **upWall**, són $\frac{\tau_\theta}{\rho} = 5.5222 \cdot 10^{-4} \text{ m}^2/\text{s}^2$. Si es substitueixen les dades a la fórmula dels esforços tangencials:

$$\tau_{\theta r} = 2 \cdot \mu \frac{\omega \cdot R_0 \cdot R_0}{R_0^2 - R_i^2} = 2 \cdot 0.1 \frac{10 \cdot 0.01 \cdot 0.01}{0.01^2 - 0.02^2} = -0.66666667 Pa \rightarrow \frac{\tau_{\theta r}}{\rho} = -0.00066667 m^2/s^2. \quad (10.39)$$

Si es calcula l'error, apareix un error del 17.16%.

11. Capítol 3: Estudi d'un flux per un colze

En aquest capítol s'aprendrà a crear la geometria amb Salome i exportar-la per tal de poder mallar-la amb Openfoam amb la funció snappyHexMesh.

11.1. Cas1: Estudi d'un colze simple

11.1.1 Proposta del problema

La proposta d'aquest problema és l'estudi d'un colze on li entra un flux uniforme de 5 m/s i al final de la canonada té una pressió de 0Pa.

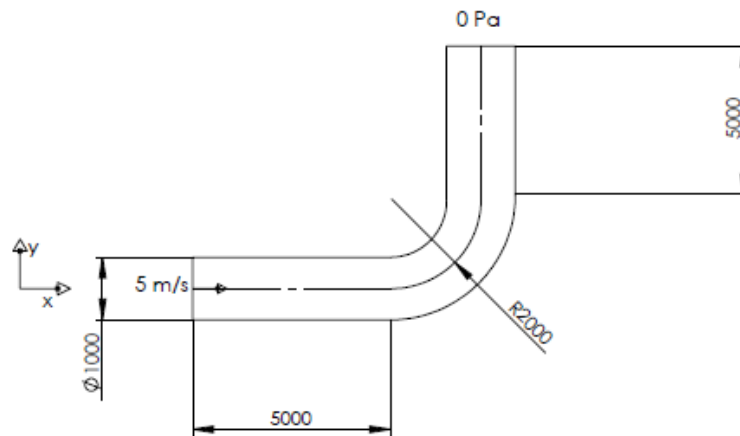


Figura 11.1: Dades per resoldre el problema. Les dades geomètriques en mm

Es farà l'estudi tant per laminar i turbulent canviant la seva viscositat cinemàtica. Per turbulent $\nu = 1 \cdot 10^5 \text{ m}^2/\text{s}$ i per laminar $\nu = 0.01 \text{ m}^2/\text{s}$.

11.1.2 Generació de la geometria del colze 3D amb el programa Salome.

Primerament executar el programa, dintre de la carpeta on hi ha el Salome, executar el fitxer run_salome.bat. A continuació, el programa s'obrirà i apareixerà una pantalla en blanc com a la imatge 11.2, una vegada desplegada la finestra del Salome clicar el mòdul de geometria.

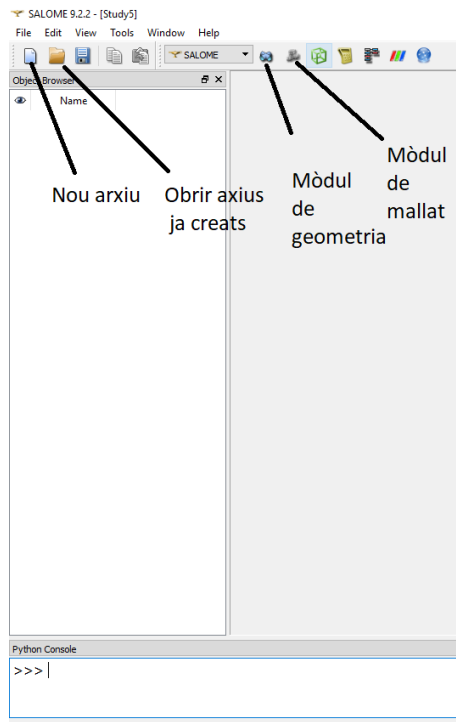


Figura 11.2: Pantalla inicial del programa Salome

Clicar l'opció 2D sketch.11.3

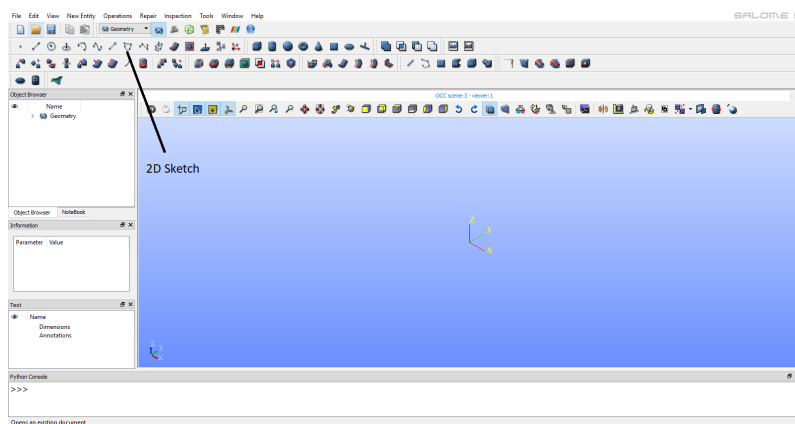


Figura 11.3: Aparat de geometria

A continuació apareixerà una finestra emergent 11.4. Es vol crear una recta per tant seleccionar l'opció de recta, pel punt (0,0) i clicar apply al punt (5,0) i clicar apply. Llavors clicar el botó d'arc de la mateixa pantalla emergent i seleccionar “direction”, l'opció de tangent de radi 2 m i un angle de 90°. Finalment clicar el botó de recta amb l'opció “direction”, “tangent” i una distància de 5 metres i per acabar clicar close.11.5

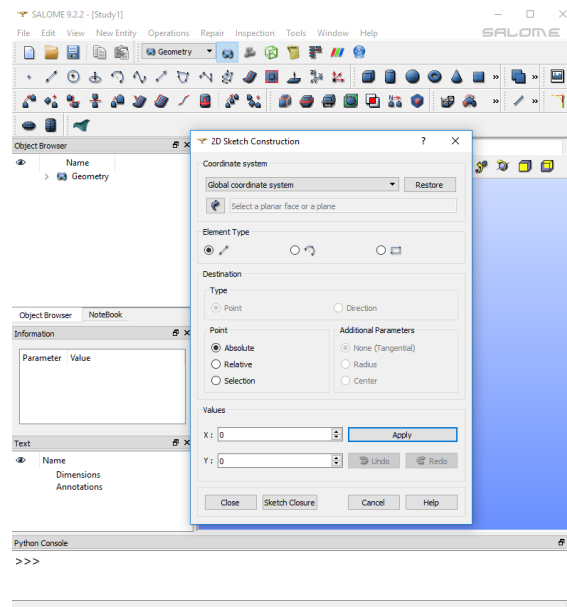


Figura 11.4: Salome amb la pantalla emergent 2D sketch

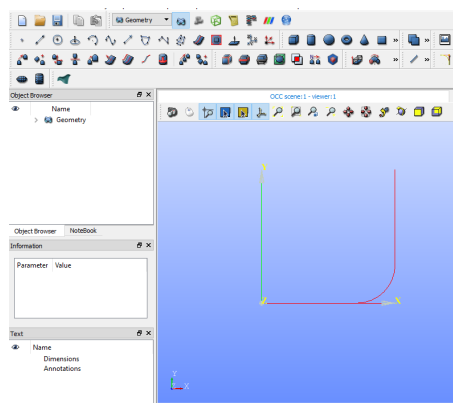


Figura 11.5: Resultat després d'aplicar les ordres del 2D sketch

Per continuar amb la generació de la geometria. És convenient crear un disc en el pla normal de l'eix de les x de 0.5 metres centrat al punt (0,0,0).11.6

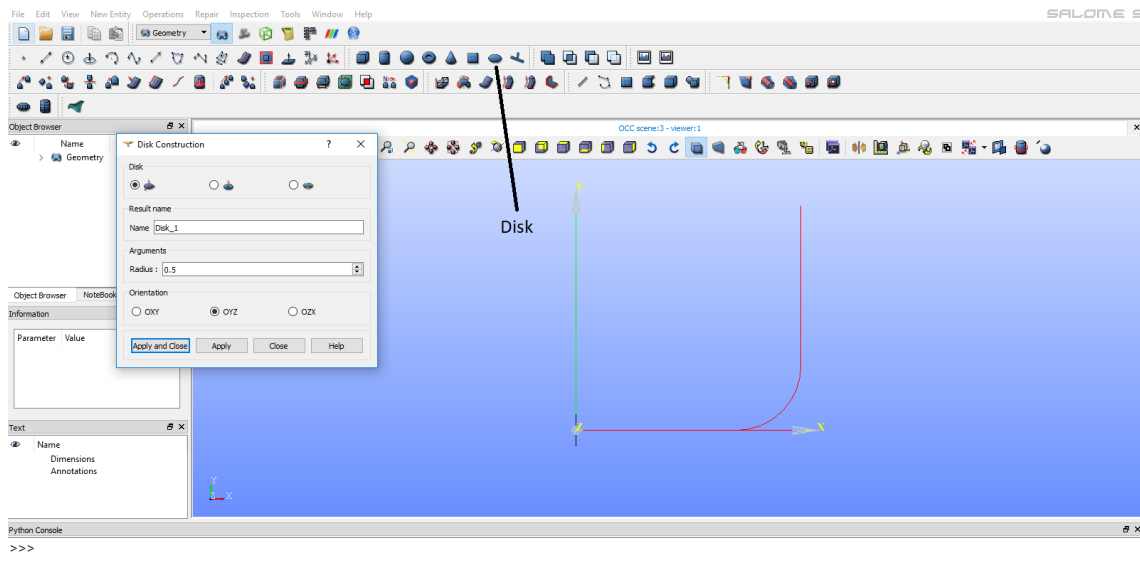


Figura 11.6: Generació del disc

A continuació s'extruirà el disc seguint el camí generat anteriorment, la funció és "Extrusion along path". A la pantalla emergent de l'extrusió cal clicar com a camí generat com a "path object" també, la circumferència com a "base object" com es pot veure a la figura 11.7.

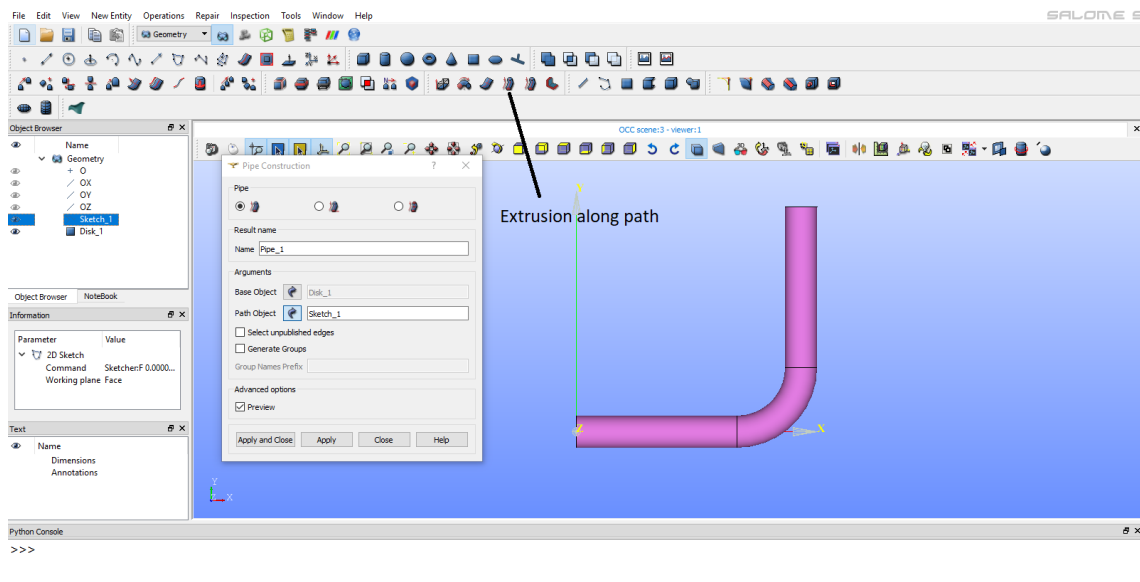


Figura 11.7: Extrusió d'un disc per un camí

Ara es pot observar que la geometria està acabada, cal diferenciar entre les diferents zones. Utilitzant la funció explode serveix per dividir la geometria en diferents cares i crearà 5 noves cares, però estaran anomenades com a "Face.1", per ordre. Per comprovar quina cara és, simplement cal fer doble clic damunt el Face1 i apareixerà d'un altre color a la geometria. Apareix a la figura 11.8

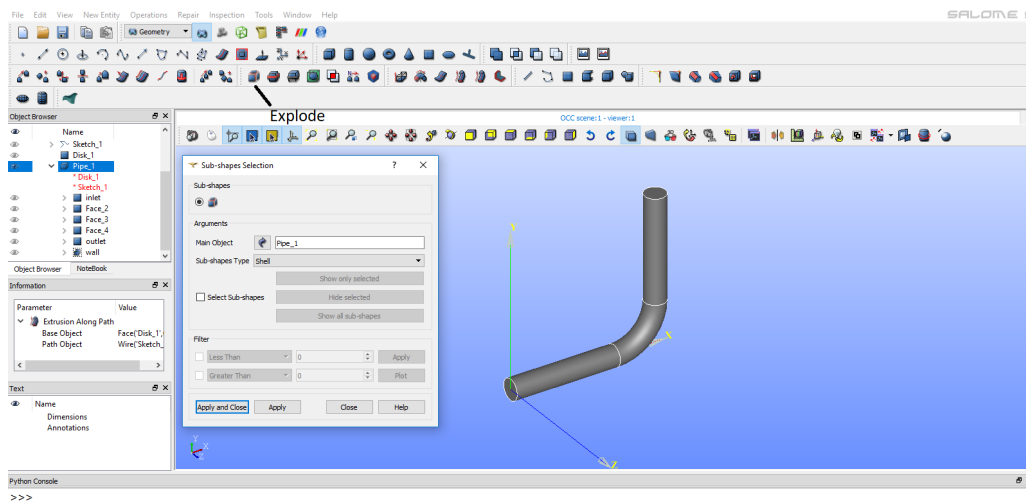


Figura 11.8: Funció explode

Per tant, una vegada identificat s'ha de canviar el nom generat pel programa. També s'ha de canviar el nom de la cara inlet i el de l'outlet. Per canviar el nom s'ha de clicar amb el botó dret on apareixerà el nom de "rename". Una vegada fet, es crearà un grup anomenat wall, per això cal, seleccionar la geometria completa a l'imatge apareix com "pipe.1", clicant una vegada, a continuació clicar el botó dret i seleccionar l'opció de create a group tal com es veu a la imatge següent 11.9, canviant el nom de Group.1 per wall. Llavors eliminar les cares inlet i outlet seleccionant-les i clicant a "remove". Una vegada es tenen totes les parets de la canonada clicar "apply and close".

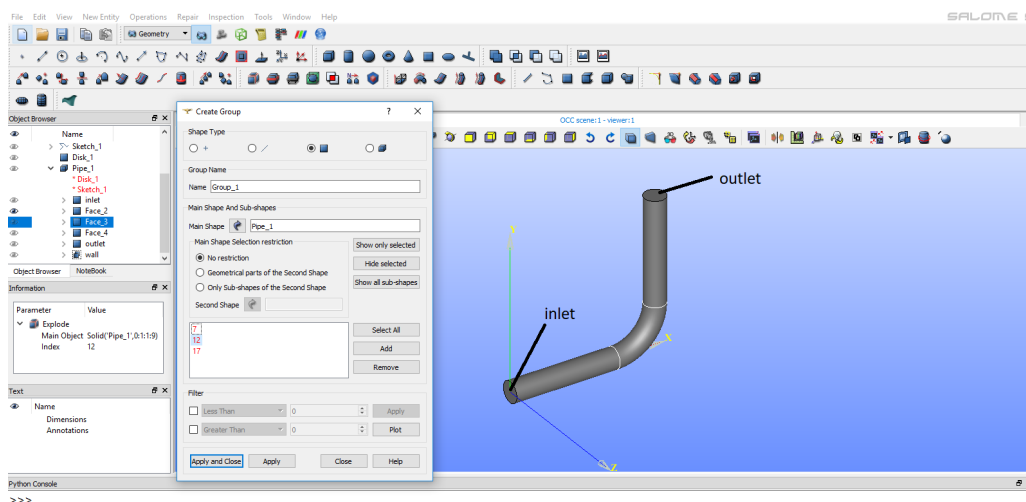


Figura 11.9: Definició de grups

Una vegada s'han creat les tres cares (inlet, outlet, wall), s'exportaran en format STL, l'opció d'importar està dintre de la carpeta "file" → export → STL. Això s'ha de fer per cada una de les cares seleccionades anteriorment. És convenient també guardar l'arxiu complet per seguretat. L'arxiu es guarda amb una extensió ".hdf".

11.1.3 Creació de la malla amb snappyHexMesh

Amb la imatge de Linux, es situarà en el directori de la carpeta on hi ha tots els casos que en capítols anteriors s'ha anomenat CFD. El que es pretén, és analitzar el colze que s'ha generat anteriorment amb una velocitat d'entrada (inlet) de 5 m/s. Una pressió a la sortida (outlet) de 0 Pa. Per tal de simplificar-ho s'agafarà un model turbulent ja creat a un tutorial d'Openfoam. Per tant s'exportarà el tutorial pitzDaily [14].

/opt/openfoam6/tutorials/incompressible/simpleFoam/pitzDaily

1

A continuació des del Windows es pot canviar el nom (per exemple, elbow). A continuació s'obrirà el programa gedit i s'obrirà els arxius “.STL” creats a l'apartat anterior. Llavors dintre del codi a la primera línia i a l'última, després de ”solid” i ”endsolid” respectivament s'afegirà el nom de la regió a la imatge 11.10 es veu representat un exemple, també s'ha de fer el mateix per outlet i wall.

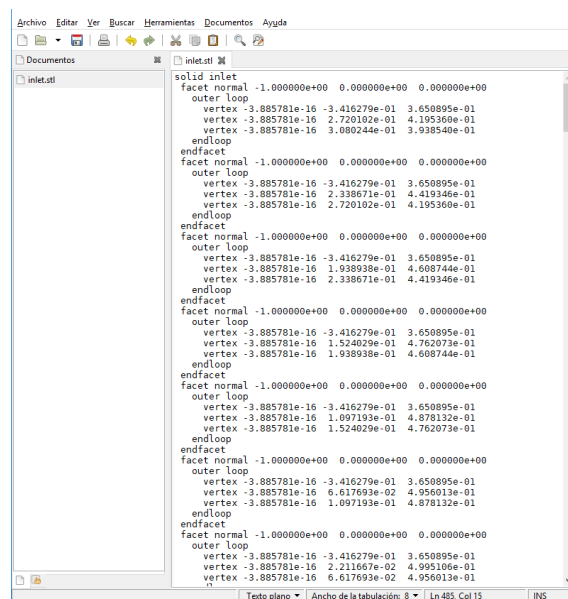


Figura 11.10: Canvi a generar dintre de l'arxiu

A continuació aquests tres arxius s'han de posar dintre d'una carpeta anomenada ”triSurface” que anirà dintre de la carpeta **constant** del cas que s'està tractant. Llavors, es procedirà a canviar aquests arxius “.stl” amb un format que ocupa menys espai que es el format “.obj”. Per tant a partir de la imatge de linux, amb el directori dintre de la carpeta “triSurface” es procedirà a utilitzar les diferents comandes:

```
surfaceConvert -clean inlet.stl inlet.obj
```

1

```
surfaceConvert -clean outlet.stl outlet.obj
```

2

```
surfaceConvert -clean wall.stl wall.obj
```

3

Finalment, s'ajuntaran totes les superfícies, amb cares “etiquetades”.

```
surfaceAdd inlet.obj outlet.obj inletOutlet.obj
```

1

```
surfaceAdd walls.obj inletOutlet.obj pipe.obj
```

2

Una vegada creat un sòlid a partir de totes les regions esborraran els arxius que no són d'interès.

```
rm *.stl inletOutlet.obj
```

1

11.1.3.1 Mallat de blockMesh

En aquest capítol es mallarà amb una funció anomenada snappyHexMesh. Però abans es crearà el que se'n diu una malla “background” amb la funció utilitzada anteriorment. El blockMesh del cas pitzDaily és massa complicada per tant no ens interessa. El que interessa és una malla que dintre hi hagi la geometria creada. Per tant es crearà un hexaedre que dintre hi hagi el colze. Per tant, l'hexaedre tindrà dimensions superiors. El codi apareix a l'**annex 1.3.1.1.1.1**.

Es realitzarà la malla i s'ha de comprovar que el colze quedi dintre. Per realitzar la malla:

```
blockMesh
```

1

```
parafoam -touch -builtin
```

2

Obrir l'arxiu “.foam” creat, des del programa paraView i a continuació obrir l'arxiu “pipe.obj” per poder saber si l'objecte està dins la malla. Com es pot observar a la imatge 11.11, a part es pot comprovar si hi ha errors a la superfície utilitzant:

```
surfaceCheck pipe.obj
```

1

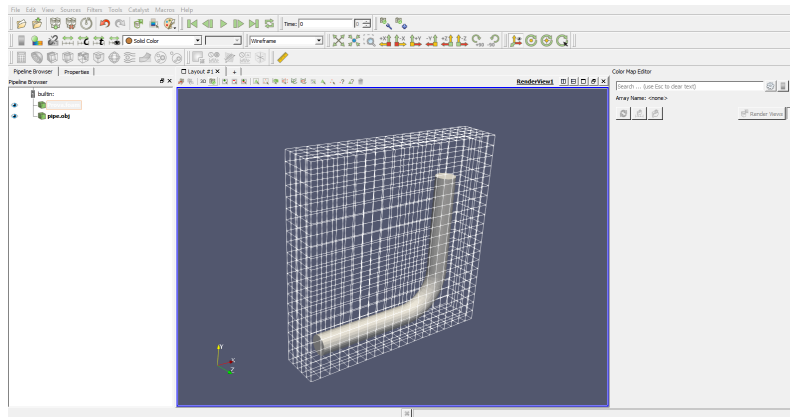


Figura 11.11: El sòlid creat i la malla del blockMesh

Ara primer, cal situar-se dintre del directori del cas, llavors es copiarà un arxiu anomenat “snappyHexMeshDict” dintre de la carpeta “system” del cas el qual s’està realitzant i s’adaptarà a la geometria.

```
cp -r /opt/openfoam6/etc/caseDicts/mesh/generation/snappyHexMeshDict system
cp -r /opt/openfoam6/etc/caseDicts/mesh/generation/meshQualityDict system
```

El fitxer del snappyHexMeshDict també es pot extreure del tutorial motorbike, que és un exemple que es troba dins els tutorials d’Openfoam, en aquest s’inclouen diferents explicacions de cada paràmetre. El directori és [14]:

/opt/openfoam6/tutorials/incompressible/simpleFoam/motorBike/ 1

11.1.3.1.1 SnappyHexMeshDict

Una vegada copiat el fitxer s’ha d’adaptar al cas. Dintre del fitxer es poden controlar una gran varietat de paràmetres com el nivell del mallat per cada àrea, el nom de les zones, es pot incloure zones de refinament de Regions, declarar el tipus de regions, si únicament són una regió (patch) o són una funció de la paret (wall). El més important a destacar és la funció locationInMesh, aquesta a partir del punt determina si es vol mallar dintre (flux intern) o fora (flux extern). Si, no es té clar on hi ha un punt dintre de l’objecte creat i fora es pot determinar el punt amb el paraview entrant a Sources → geometric shapes → Sphere, en aquest apartat es pot anar provant punt a punt on hi ha una esfera.

Hi ha indicacions de cada paràmetre de com s’ha de programar. Es pot consultar en el mateix **annex 1.3.1.1.6**. Els noms de les cares queda definit en el mateix arxiu

”snappyHexMeshDict”. Els noms de les cares són inlet, outlet i wall (els mateixos que generats en el Salome).

11.1.3.1.2 MeshQualityDict i SurfaceFeatureExtractDict

Aquests dos arxius complementen l'arxiu ”snappyHexMesh” per tal de tenir millors resultats, els dos també van dintre de system. [8]

S'ha de modificar l'arxiu meshQualityDict, però únicament s'ha de modificar perquè llegeixi el fitxer adequat. Aquest fitxer fa referència a controls de qualitat de la malla. També es pot extreure del tutorial motorbike. Es pot consultar a l'**annex 1.3.1.1.5**.

A partir d'aquí ja es podria executar, des del directori del cas, indicant:

```
snappyHexMesh
```

1

Una vegada acabat si es genera un arxiu ”.foam”, es pot visualitzar:

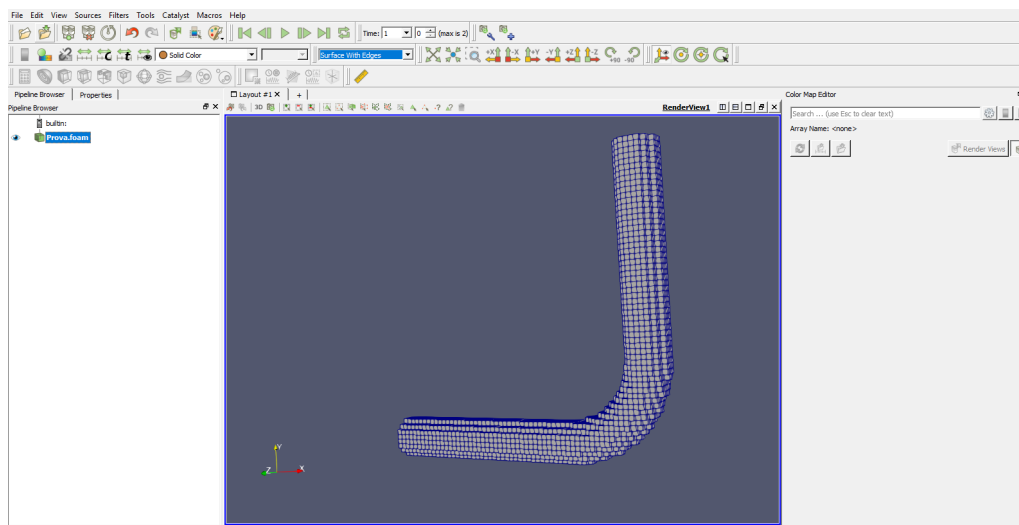


Figura 11.12: Malla executada per snappyHexMesh

Com es pot veure la malla no té ben extretes les superfícies, també es podria solucionar augmentat el nombre de divisions en el blockMesh. Però el que es fa per definir-ho millor és incloure un fitxer anomenat ”surfaceFeatureExtractDict”, que es pot trobar al tutorial motorbike. Simplement cal canviar el fitxer d'obertura. En cas de dubte es pot consultar a l'**annex 1.3.1.1.7**. Com es pot veure ara en el cas s'han creat dues carpetes anomenades 1 y 2 (temps segons system/controlDict). En cada carpeta hi ha un resultat parcial de cada fase castellated y snap). Cal esborrar les dues. Una vegada inclòs el fitxer es procedirà a sobreesciure els resultats de la malla utilitzant:

snappyHexMesh –overwrite

1

I els resultats són els següents:

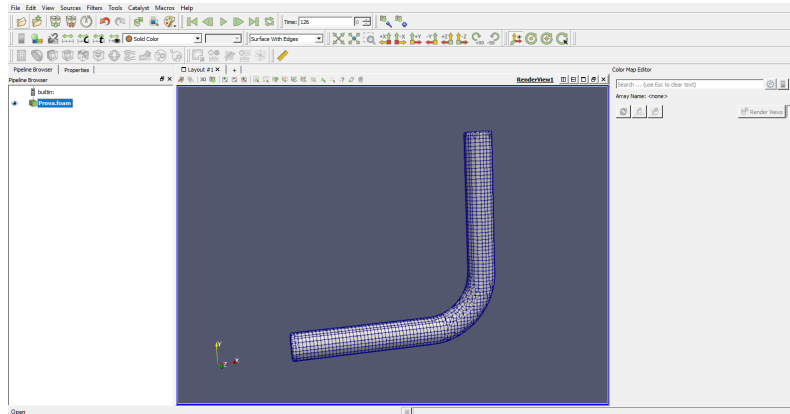


Figura 11.13: Malla amb l'arxiu "surfaceFeatureExtractDict"

Es pot dir que la malla ja està generada.

11.1.4 Resolució amb un model turbulent

En aquest apartat s'assumirà que:

- Flux turbulent.
- Fluid incompressible.
- Fluid newtonià.
- Es negligiran les forces gravitatòries.
- Flux estacionari.

Condicions que té el solver SimpleFoam.

11.1.4.1 Preparació del codi

Primerament, cal comprovar si el cas és turbulent. Per calcular el nombre de reynolds, se suposarà una viscositat cinemàtica de $1 \cdot 10^{-5}$. La velocitat d'entrada és de 5 m/s i el

diàmetre és de 1m:

$$Re = \frac{v_x \cdot D}{\nu} = \frac{5 \cdot 1}{1 \cdot 10^{-5}} = 500000 \quad (11.1)$$

Per tant s'haurà d'utilitzar un solucionador turbulent, en aquest cas s'utilitzarà el solver simpleFoam. El model s'ha extret del tutorial pitzDaily. Simplement s'ha de fer els canvis oportuns que s'indiquen a continuació:

- Carpeta 0: Cal fer el canvi del nom de les regions i eliminar la regió sobrant de cada variable. En els casos concrets que s'indiquen a continuació, són per adaptar el problema amb les condicions establertes anteriorment. En cas de dubte es pot consultar a l'**annex 1.3.1.1.3**.

► Carpeta U: **inlet** s'ha de definir amb un flux constant de 5m/s (fixedValue i amb valor (5 0 0)), **wall** es pot declarar amb una funció anomenada noSlip que és el mateix que tenir fixedValue amb valor (0 0 0) i en el cas de l'**outlet** zeroGradient, ja que no es sap el valor. Més concretat apareix a l'**annex 1.3.1.1.3.8**.

► Carpeta p: La pressió de sortida (**outlet**) a amb la funció fixedValue amb valor 0, les altres dues cares zeroGradient. Apareix a l'**annex 1.3.1.1.3.7**.

- Carpeta system: Dintre de system, es pot canviar el nombre d'iteracions canviant el valor d'endTime ve predefinit a 4000 iteracions, s'afegirà dintre de "functions". La funció per representar els resultats i la del càlcul d'esforços tangencials (per aprofitar que simpleFoam sí que permet calcular esforços tangencials.). Els arxius de system es troben dintre de l'**annex 1.3.1.1.1**.

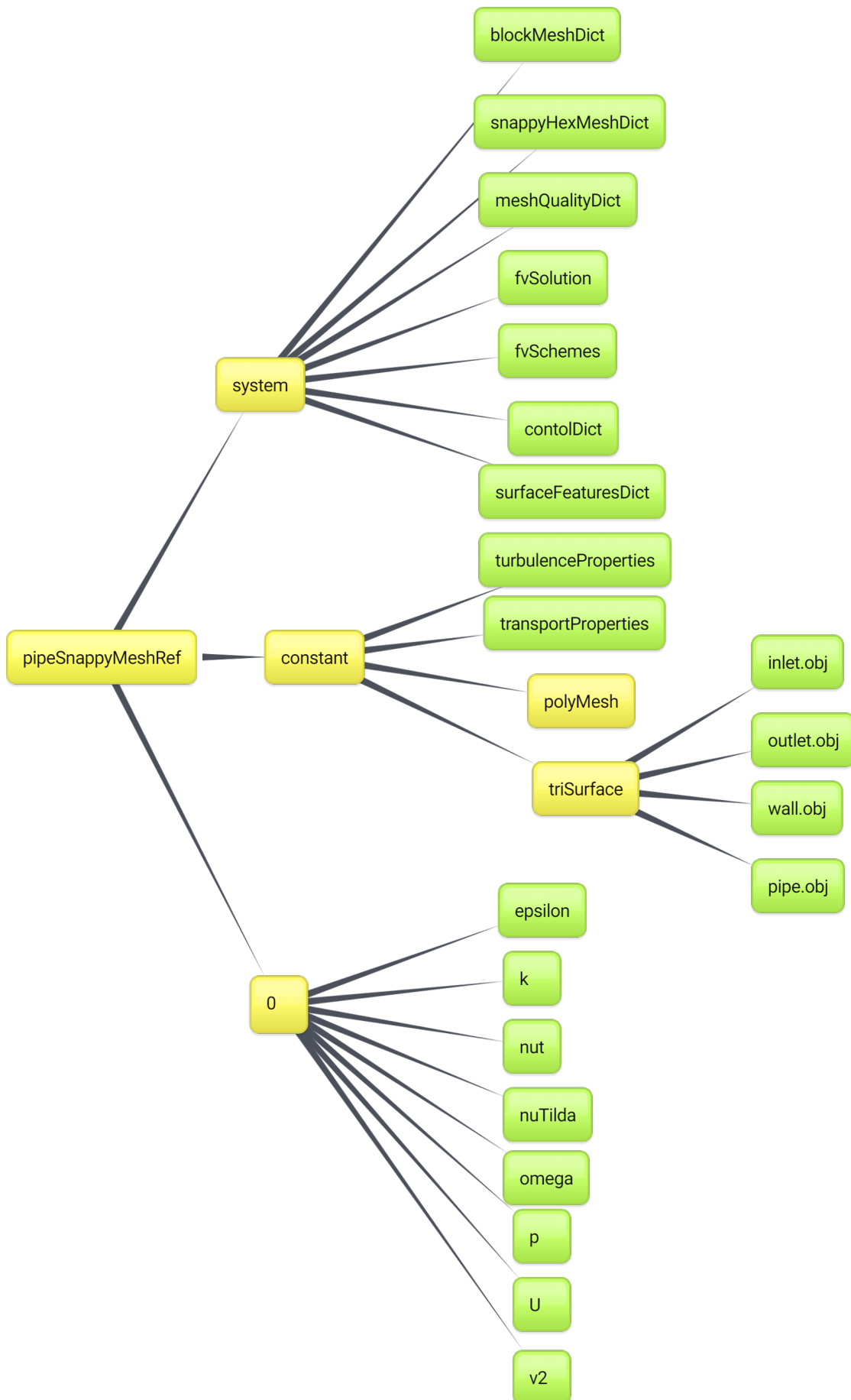
```

functions                                     1
{                                              2
    #includeFunc residuals                    3
    #includeFunc wallShearStress              4
                                              5
}                                              6

```

- Constant: assegurar-se que la viscositat cinemàtica té el valor de $1 \cdot 10^{-05}$. El codi sencer de la carpeta constant es pot consultar a l'**annex 1.3.1.1.2**.

L'esquema del cas és el següent:



11.1.4.2 Execució del cas

Una vegada generada la malla i adaptant les condicions de contorn, és hora d'executar la solució. Cal observar que al generar la funció `snappyHexMesh` s'han creat dues carpetes relacionades als intervals de temps. La primera carpeta representaria al primer "time step" fa referència a la malla de la figura 11.12, generada per la funció `snappyHexMesh`. La segona carpeta fa referència al segon "time step" que una vegada és generada es pot observar que és més precisa i fa referència a la figura 11.13, aquesta nova malla és generada gràcies a l'arxiu "surfaceFeaturesDict". Per tant cal canviar el nom de la carpeta 0 per 0.org. A continuació copiar el que hi ha dintre de la nova carpeta 0.org i posar els fitxers (les condicions de contorn) dintre la carpeta que tengui el major "timestep" (la carpeta que conté la malla més precisa) i canviar el fitxer "controlDict" per tal que comenci a llegir la carpeta amb la malla més precisa.

Una altra forma de realitzar aquesta operació sense haver de tocar l'arxiu "controlDict" és canviar el nom de la carpeta 0 per la de 0.org. Però a continuació canviar el nom de la carpeta del valor més alt per al nom de 0, esborrar la carpeta intermèdia i dintre de la nova carpeta 0 afegir els arxius de la carpeta 0.org. Llavors es pot introduir la comanda: [4]

```
simpleFoam
```

1

Una vegada executat tot es pot veure que ha arribat a les interaccions màximes que se li han indicat al programa això vol dir que la solució no ha convergit. Indicant aquestes ordres a la imatge de Linux en el directori del cas:

```
rm -rf postProcessing
```

1

```
simpleFoam > log &
```

2

Fer doble clic a Xming per activar-lo es procedeix a:

```
foamMonitor -l postProcessing/residuals/0/residuals.dat
```

1

Els resultats són els següents:

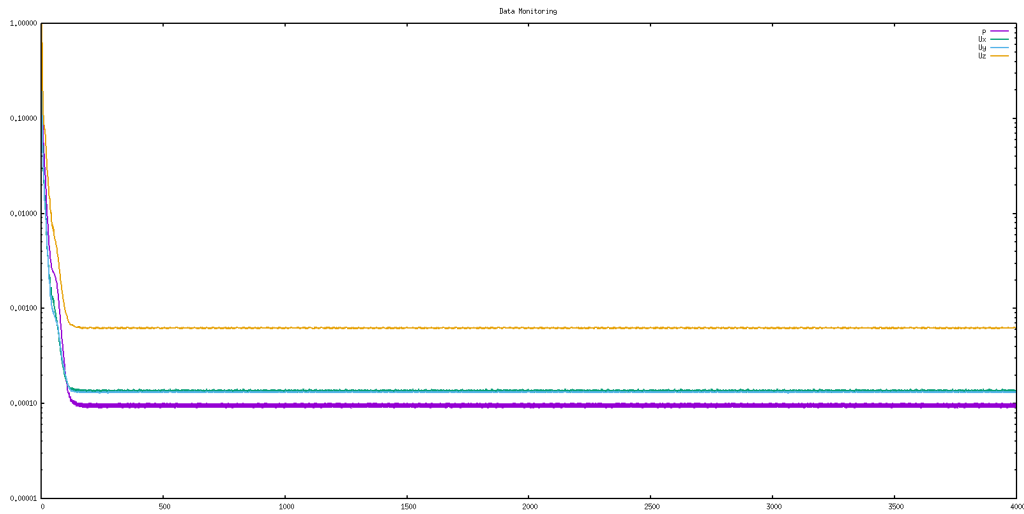


Figura 11.15

Com es pot veure a la gràfica la magnitud U_z queda un residual molt alt i no disminueix. Això fa que el programa no convergeixi amb les interaccions donades. Per tant una solució és posar un factor de relaxació més baix, aquest cas duu un factor de relaxació d'un 0.9, per tant es procedirà a baixar-lo. Els factors de relaxació en aquest cas, es poden modificar a l'arxiu fvSolution on posa "relaxing factor", es farà la prova amb un factor de relaxació del 0.7. Es procedirà a esborrar les carpetes generades per les iteracions anteriors i tornar a executar el programa. Una vegada s'ha tornat a executar l'ordre **simpleFoam**. Es torna a representar els residuals i els resultats obtinguts són els següents [10] :

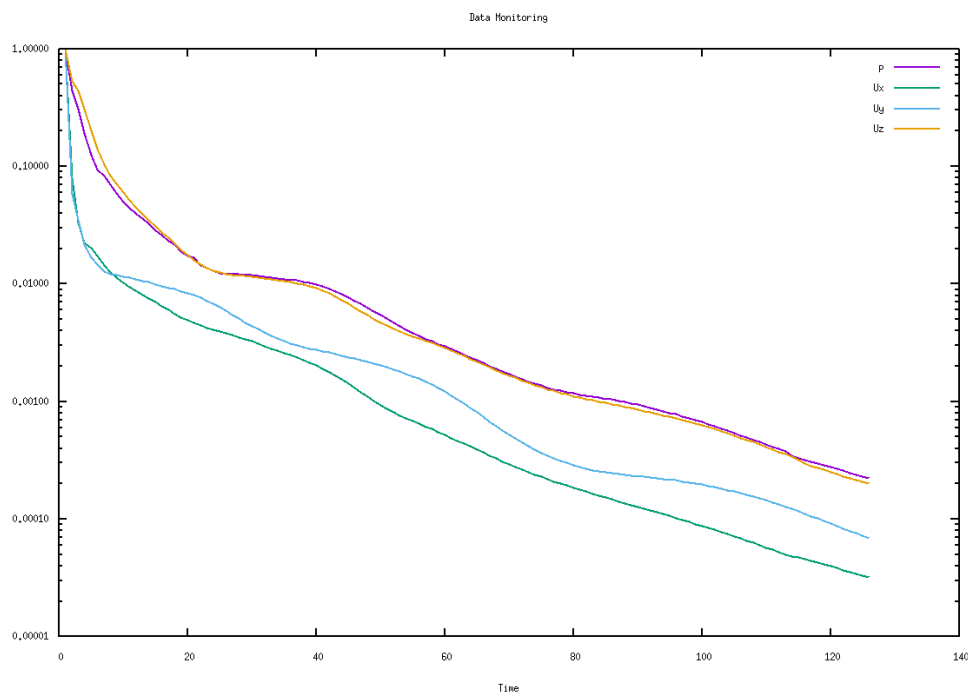


Figura 11.16: Gràfica dels residuals aplicant un factor de relaxació

Com es pot observar en aquest cas, els resultats baixen i U_z acaba tinguent un residual més baix que a l'anterior cas i acaba convergint a la iteració 126.

11.1.4.2.1 Tall i seccions per la visualització de resultats

Els resultats són els següents, en aquest cas per poder visualitzar el camp de velocitats s'ha fet un tall amb l'eina del paraview clip, una vegada clicat s'ha de seleccionar el tall corresponent, en aquest cas s'ha seleccionat "z normal" genera un pla normal a l'eix z com es pot veure a la imatge 11.17, també es podria visualitzar amb l'eina del costat anomenada "slice", la diferència clau és que "clip" genera un tall mentre que "slice" genera una secció.

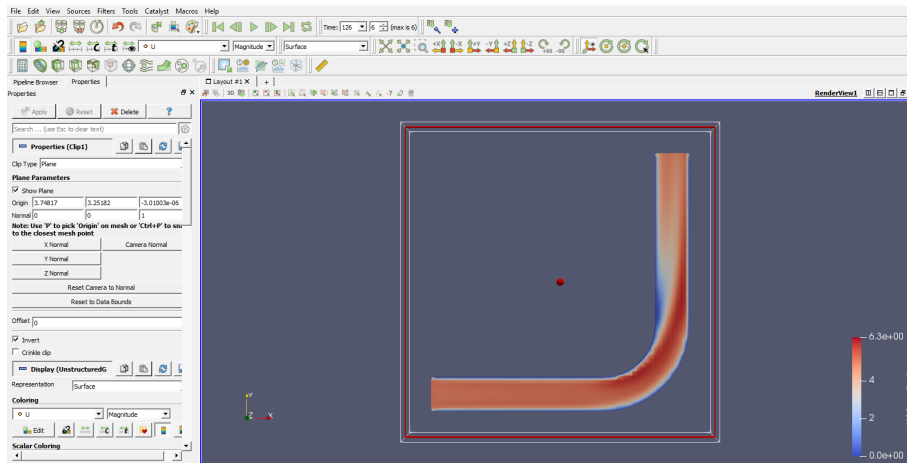


Figura 11.17: Resultats de la velocitat del colze simple amb un tall amb l'eina "clip"

11.1.4.2.2 Determinar la mitjana d'una unitat a una regió concreta

A vegades és interessant determinar la pressió a una secció. Una forma un poc més elaborada que s'ha fet anteriorment seria generar una gràfica damunt la regió a partir d'una recta, veure com és aquesta funció i quin ordre de magnitud té. Però a vegades simplement es vol tenir una idea de quina és la pressió mitjana o també comprovar que la pressió que s'ha establert predefinida d'aquella regió és la correcta una forma fàcil i ràpida és utilitzant.

```
postProcess -func 'patchAverage(name=outlet ,p)'
```

1

Com es pot veure a la regió outlet s'ha predefinit a un valor de $0 \text{ m}^2/\text{s}^2$ per tant si s'executa la funció hauria de sortir també 0:

```

Time = 126

Reading fields:
  volScalarFields: p

Executing functionObjects
surfaceFieldValue patchAverage(name=outlet,p) write:
  areaAverage(outlet) of p = 0

End

```

Figura 11.18: Pressió a la regió outlet

I en el cas de la regió inlet, es desconeix el valor de la pressió mitja:

```

Time = 126

Reading fields:
  volScalarFields: p

Executing functionObjects
surfaceFieldValue patchAverage(name=inlet,p) write:
  areaAverage(inlet) of p = 3.46459

End

```

Figura 11.19: Pressió a la regió inlet

Per tant es pot observar que la regió inlet té una pressió mitjana de $3.46459 \text{ m}^2/\text{s}^2$. Per tant per un flux de 5 m/s es necessita una pressió de 3465 Pa a la regió inlet (suposant una $\rho = 1000 \text{ kg/m}^3$).

11.1.5 Resolució amb un model laminar

Per un model laminar solucionat amb el solver IcoFoam. S'assumeix:

- Fluxe laminar.
- Fluid incompressible.
- Fluid newtonià.
- Es negligiran les forces gravitatòries.
- Flux estacionari.

Dintre d'aquest cas simplement, es deixarà l'esquema que s'ha de seguir. Així i tot, l'estructura es pot importar del cas d'Openfoam. Des del directori de la carpeta dels casos. Des del directori principal, introduir la següent comanda:

```
cp -r $FOAM.TUTORIALS/incompressible/icoFoam/elbow CFD
```

1

Per adaptar-lo al cas:

- Canviar el nom per no provocar confusió, per exemple, pipeSnappyMeshRefIcoFoam.
- Esborrar l'arxiu ".msh".
- Canviar el valor de ν per un nombre que sigui laminar, el valor proposat: 0.01.
- Adaptar els arxius de la carpeta 0, U i p.
- Generar la malla de la mateixa forma que l'apartat 11.1.3
- El valor deltaT (dintre de l'arxiu controlDict) proposat és 0.005 i endTime amb valor de 20.

Executant amb l'ordre icoFoam. Tot el codi es pot consultar a l'**annex 1.3.1.2**.

11.1.6 Comparació dels del cas turbulent amb el cas laminar amb paraview

A vegades és necessari comparar diferents resultats dintre de la mateixa pantalla del paraview, per visualitzar de forma més clara les diferències i poder comparar els resultats. En aquest apartat s'explicarà com s'ha de fer. A més en aquest apartat es compararan els camps de velocitats.

- Obrir els dos arxius.
- Clicar damunt l'arxiu del cas laminar de forma que quedi seleccionat i entrar dins la pestanya "Properties" i a l'apartat "transforming" dintre de "translation", la primera columna representa la x la segona la y i tercera la z, si es posa, el nombre 10 en la primera columna es desplaçarà la geometria del cas seleccionat.
- A continuació es farà un tall. No es pot realitzar un tall pels dos sinó que s'ha de fer per separat. Seleccionar un dels dos casos simplement clicant una vegada i clicant damunt "clip" com s'ha fet a l'apartat anterior i el tall perpendicular a l'eix de la z i clicant apply. Assegurar-se de què es visualitza la velocitat.

- Repetir el mateix però seleccionant l'altre cas. Encara que el tall pareixi que el fa damunt el cas de l'altre clicar "apply" i canviar a 10 de l'apartat de "translation".

El resultat esperat és aquest. Fixar-se amb l'esquema que apareix a l'esquerra de la imatge.11.20

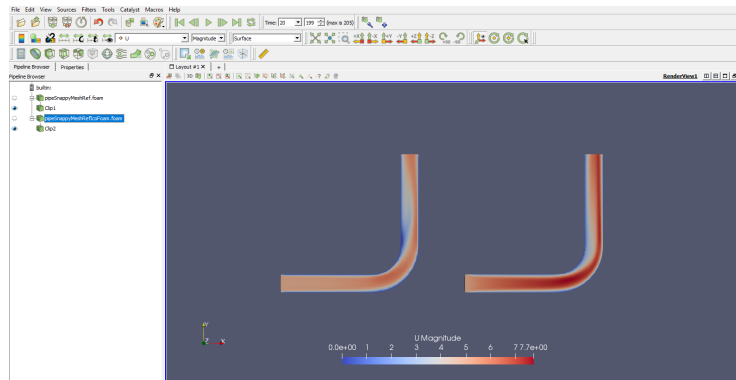


Figura 11.20: A l'esquerra apareix el cas turbulent i a la dreta apareix el cas laminar amb una mateixa escala

11.2. Cas 2: Estudi d'un colze amb dues entrades

11.2.1 Proposta del problema

En aquest cas s'afegirà una entrada tangent a l'entrada del colze. Amb les mateixes condicions anteriors simplement es canvia el valor d'inlet1 per 2 m/s i a inlet2 per 5m/s.

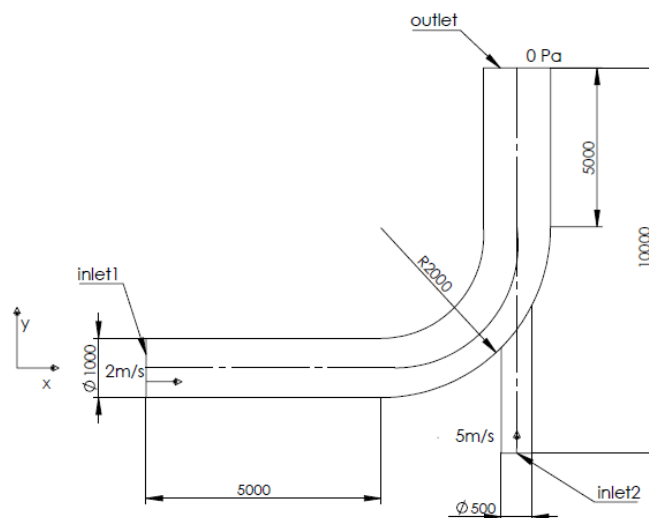


Figura 11.21: Colze amb dues entrades, les dades geomètriques estan en mm.

Es farà l'estudi per un flux laminar amb una viscositat cinemàtica de $\nu = 0.01 \text{m}^2/\text{s}$.

11.2.2 Generació de la geometria

En aquest cas es partirà de la geometria d'un colze simple generat anteriorment, per tal de crear una segona entrada. És convenient, copiar l'arxiu i canviar el nom.

Per tant, es procedirà a obrir el programa salome. Obrir el cas guardat anteriorment ".hdf". Esborrar les cares generades per la funció "explode" clicant amb el botó dret damunt cada cara de l'esquema que genera a la dreta del programa i clicar damunt remove.

Una vegada fet això es procedirà a generar un punt damunt una superfície. Clicant damunt l'opció de punt marcat a la imatge. Sortirà una pantalla emergent. El punt serà creat damunt la superfície outlet. Es pot clicar damunt de la superfície directament i una vegada seleccionats tots els paràmetres clicar "Apply and Close".

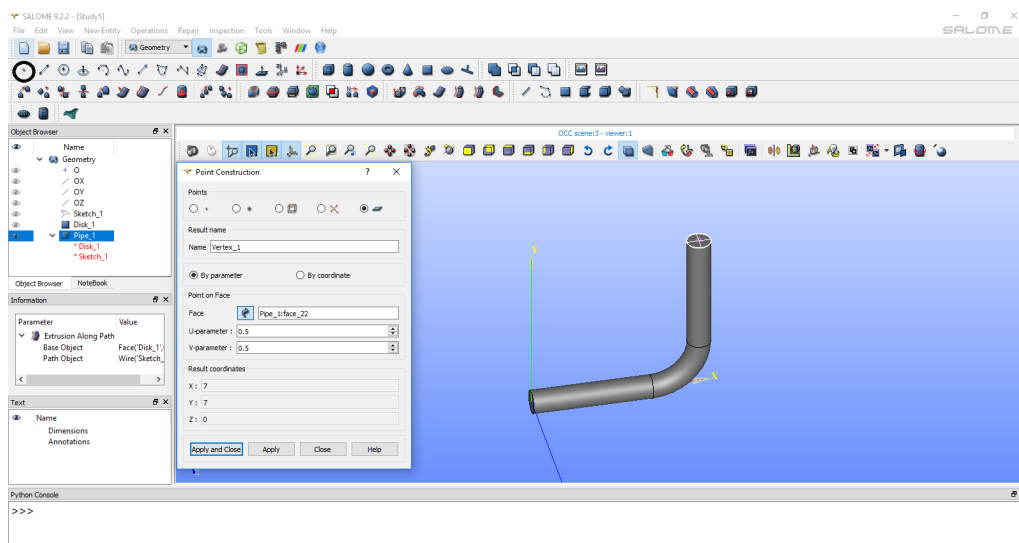


Figura 11.22: Generació d'un punt damunt una superfície.

El següent pas és generar un disc damunt la superfície outlet per tal de crear la segona entrada. Utilitzant l'opció "Disk", de radi 0.25m. De la següent forma: 11.23

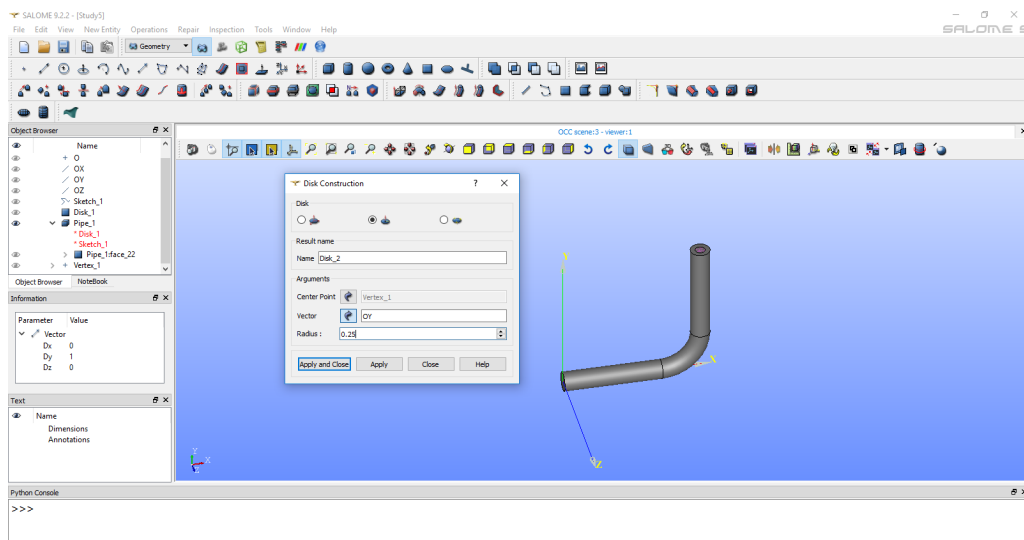


Figura 11.23: Generació del disc

Una vegada creat cal extrusionar aquest disc fins a una altura de -10m. Figura 11.24.

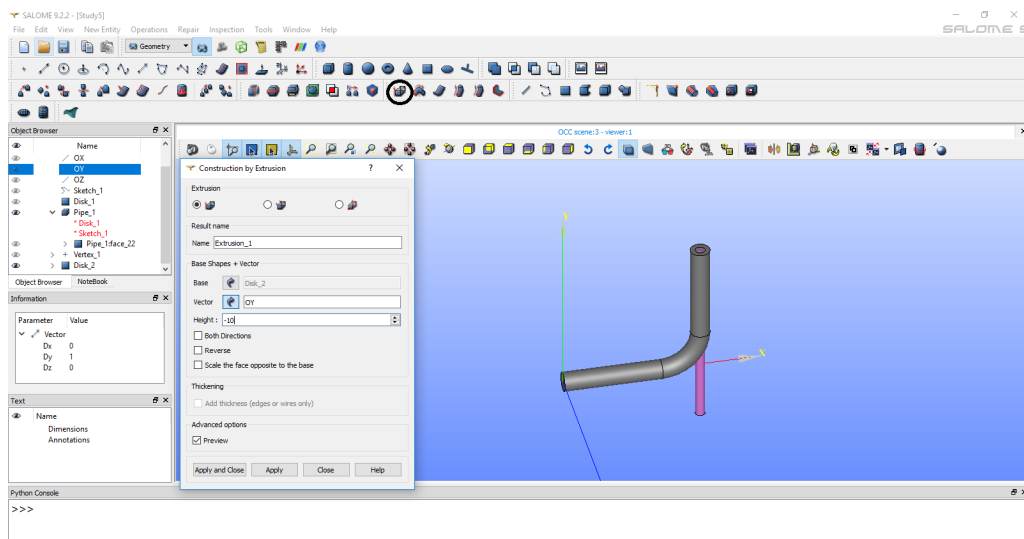


Figura 11.24: Extrusió

Ara es té creat dos objectes un incrustat dintre de l'altre per tal de formar un únic objecte simplement cal utilitzar la funció "fuse" aquesta funció unirà els dos objectes amb la finalitat de crear-ne únicament un.11.25

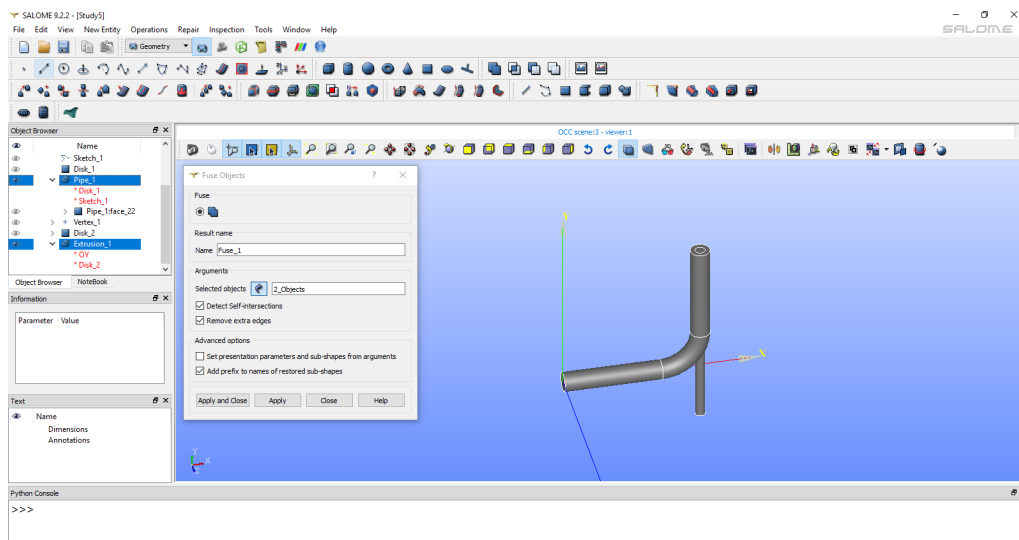


Figura 11.25: Utilització de la funció fuse

Una vegada fusionats els dos objectes, s'ha d'utilitzar la funció ja indicada en el cas anterior "explode" i finalment reanomenar les cares i crear un grup de cares com s'ha fet anteriorment, però en aquest cas:

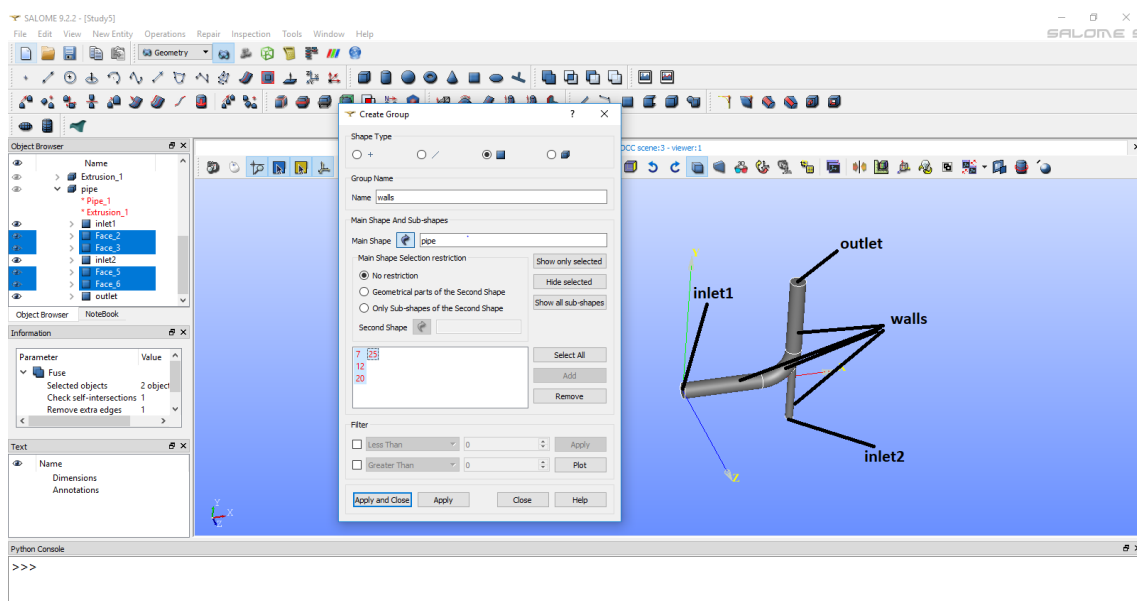


Figura 11.26: Nom de les cares i l'agrupació de cares per formar walls.

11.2.3 Creació del cas i de la malla

Aquest cas es resoldrà de forma laminar, es pot agafar el cas anterior creat i fer les modificacions pertinents:

1. Primerament obrir els arxius ".stl" amb el gedit i afegir a la paraula solid el nom de la cara (primera línia de codi) i a endSolid (última línia de codi) també, per cada cara.
2. Passar les 4 cares de ".stl" a ".obj".
3. Utilitzar la funció "Addsurface" utilitzada anteriorment.
4. Crear una malla amb el blockMesh que l'objecte estigui amb tota la seva totalitat dintre.
5. Fer les modificacions pertinents de nomenclatura dels arxius que s'utilitzen per a la funció snappyHexMesh (snappyHexMeshDict, meshQualityDict i surfaceFeaturesDict) i executar la funció snappyHexMesh. La malla té aquesta forma:

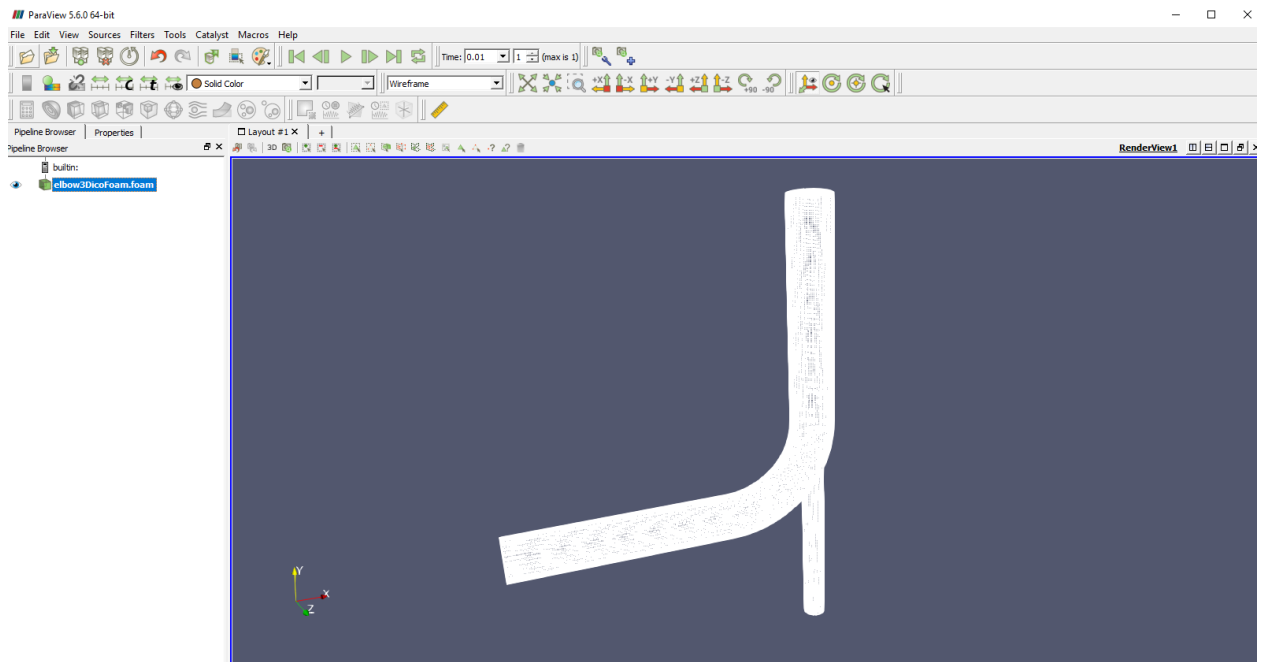


Figura 11.27: Malla del colze amb dues entrades

6. Adaptar en el cas la carpeta 0,

Pel cas del fitxer U. Es definirà inlet1, hi entrarà el fluid a 2 m/s a l'eix de les x. A inlet2 entrarà el fluid a 5m/s a l'eix de les y, outlet serà definida com a zeroGradient, finalment la regió walls serà descrita per noSlip on la velocitat a les parets serà igual a 0.

La pressió a la sortida serà igual a 0 mentre que a les altres tres serà definida com a zeroGradient.

7. La carpeta system. Si s'ha establert la malla correctament amb l'snappyHexMesh, simplement serà necessari canviar el nonOrthogonal posant un 1 i no 2, dintre del fitxer fvSolution [10].

Es pot consultar el codi sencer a l'**annex 1.3.2**.

11.2.4 Execució i solucions del cas

S'executarà el solver icoFoam per tal d'obtenir les solucions. A continuació es crearà l'arxiu ".foam" i s'obrirà el cas amb el paraview.

- Per les velocitats:

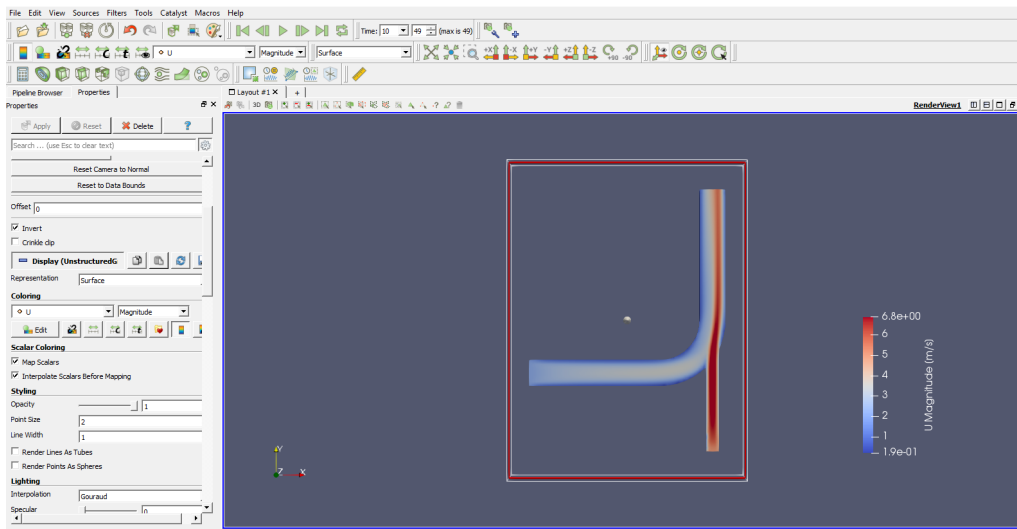


Figura 11.28: Camp de velocitats del cas en m/s

- Per les pressions:

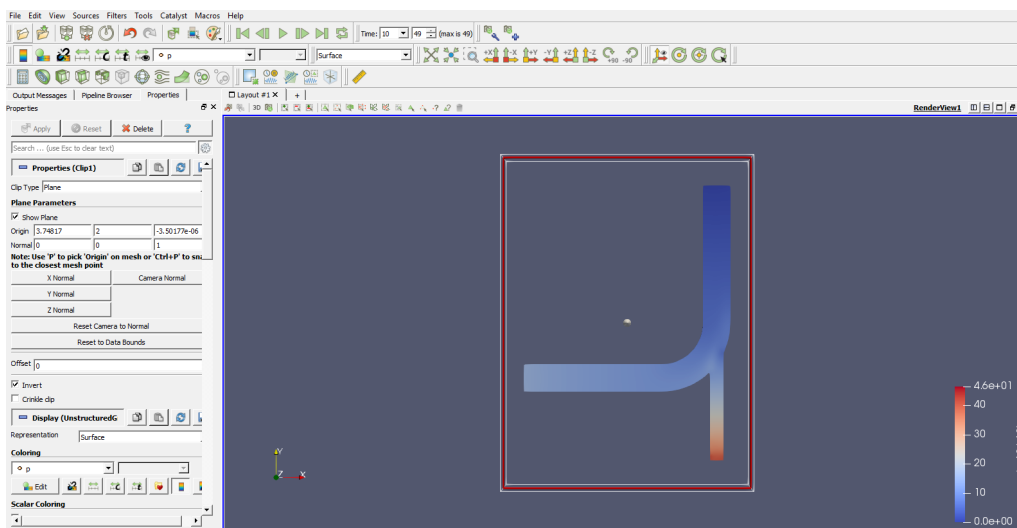


Figura 11.29: Resultat de la pressió en m^2/s^2

- Es comprovarà que el caudal que entra ha de ser el mateix que surt, ja que és incompressible s'utilitzarà per a les regions inlet1 inlet2 i outlet amb la funció:

```
postProcess -func "flowRatePatch(name=outlet)"
```

I s'obtenen els següents resultats:

► Outlet:

```
Time = 10

Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=outlet) write:
  sum(outlet) of phi = 2.34843

End
```

Figura 11.30: Cabal d'outlet en m^3/s

► inlet1:

```
Time = 10

Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=inlet1) write:
  sum(inlet1) of phi = -1.44487

End
```

Figura 11.31: Cabal inlet1 en m^3/s

► inlet2:

```
Time = 10

Reading fields:
  surfaceScalarFields: phi

Executing functionObjects
surfaceFieldValue flowRatePatch(name=inlet2) write:
  sum(inlet2) of phi = -0.903564

End
```

Figura 11.32: Cabal d'inlet2 en m^3/s

Si es fa la suma, s'obté $2.34843 - 0.90356 - 1.44487 = 0 \text{ m}^3/s$ per tant el cabal que entra

és el mateix que el que surt. És una forma ràpida de comprovar que la simulació té sentit i respecta l'equació de la continuïtat.

12. Capítol 4: Estudi d'un flux d'aire per un F1

12.1. Proposta del problema

Per tal de concloure amb la guia. Es posarà l'exemple de com crear una malla externa d'un objecte. Per tant en aquest capítol s'importarà una geometria 3D externa i es farà l'estudi. Es poden descarregar diferents geometries a la web. En aquest cas serà un F1 [25].

Es faran les següents hipòtesis:

- Flux incompressible.
- Flux turbulent.
- El cos és indeformable i els components del model 3D són estàtics.
- Flux newtonià.
- Efectes de la gravetat negligibles.

Per tal de determinar que és un flux turbulent la viscositat cinemàtica a 20°C del aire és de $\nu = 15.10 \cdot 10^{-6} m^2/s$. Es suposa una velocitat 50 m/s que serien uns 180 km/h. En aquest cas la longitud caraterística d'aquest cas serà la llargada del cotxe que és de aproximadament 5.78m. [27]

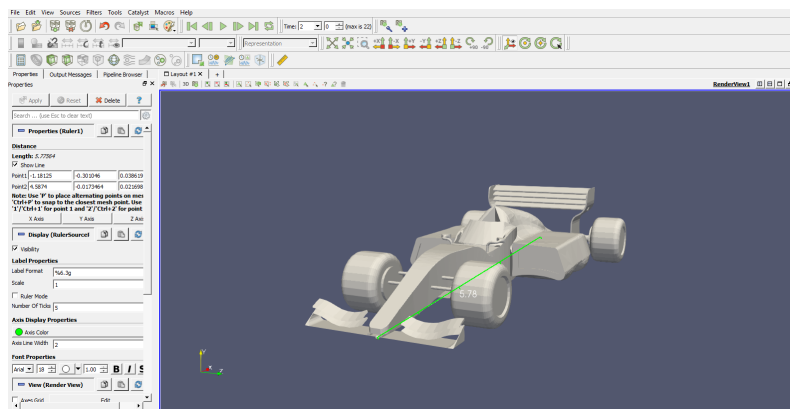


Figura 12.1: Ús de l'eina de mesura amb paraview

Per tant, es determina que la llargada és de 5.78m.

$$Re = \frac{U \cdot d}{\nu} = \frac{50 \cdot 5.78}{1.510 \cdot 10^{-5}} = 19.27 \cdot 10^6 \quad (12.1)$$

Per al valor de reynolds calculat es pot considerar que el flux és turbulent. L'esquema a seguir en aquest cas serà el mateix que en el capítol 3. Amb la diferència que a la geometria només tindrà una regió.

12.2. Adaptació de la geometria.

En aquest cas, una vegada descarregada la geometria. Cal tenir en compte que l'arxiu té una extensió ".stl". Aquesta extensió OpenFoam la detecta. Però si no fos el cas, és aconseguir la geometria amb l'extensió que pot obrir salome. Aquestes extensions que pot obrir el salome són, BREP, STEP, IGES, STL, XAO i VTK.

Per obrir la geometria amb Salome únicament cal clicar a **File** → **Import** → format en el qual es té la geometria.

Una vegada importada, cal revisar les escales i les translacions que un desitja. Ja que en importar pot conduir a errors. És convenient sempre importar-ho amb metres i després aplicar l'escala i fer les modificacions pertinents, ja que el programa pot ser que no deixi importar. Per aplicar l'escala o traslladar s'utilitzaran les següents icones.

Per tal de no haver de simplificar les operacions es recomana que sigui una mateixa geometria única i no tingui diferents peces. Això es pot aconseguir amb el programa utilitzat per generar la geometria utilitzada l'extensió ".STEP". A continuació convertir-lo amb Salome a un arxiu amb extensió ".STL".

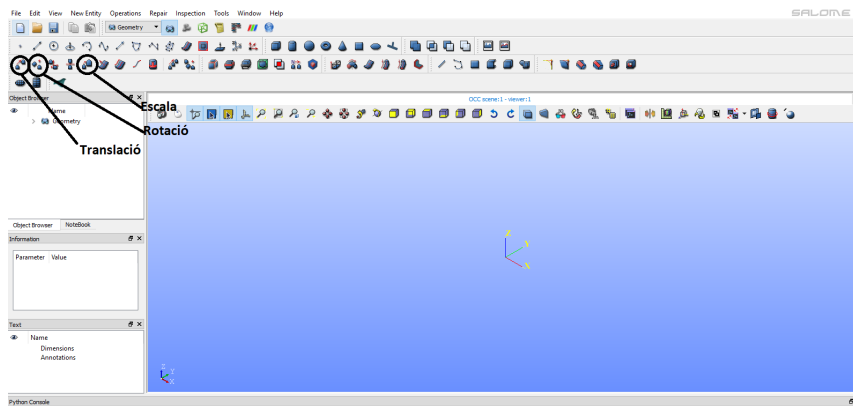


Figura 12.2: Icones per tal d'adaptar la geometria a l'usuari

La geometria que queda en aquest cas és:

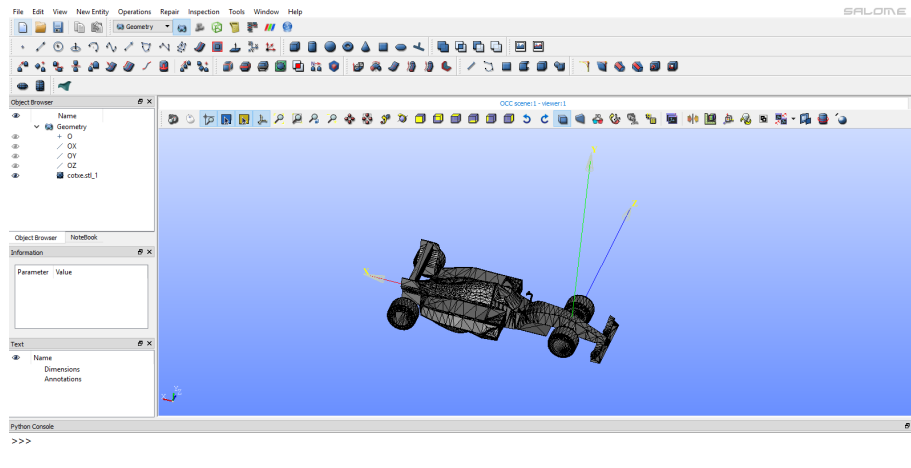


Figura 12.3: Geometria final

Una vegada s'hagin fet els canvis pertinents, simplement cal anar clicar damunt la darrera modificació i que aparegui seleccionada tota la geometria. Llavors exportar a un arxiu STL. File → Export → STL, en aquest cas s'ha guardat l'arxiu amb el nom de "cotxe.stl".

12.3. Preprocés

Per tal de procedir amb el mallat de l'objecte on serà mallat amb la funció snappy-HexMesh. Es seguirà el mateix procediment que en el capítol 3 però en aquest cas només es té una superfície, per tant no caldrà accedir dintre dels arxius per tal d'anomenar-los. Abans de res es convertirà l'arxiu de la geometria de ".STL" a ".obj" amb Linux i dintre el directori dintre de la carpeta "TriSurface" s'executarà la següent funció:

```
surfaceConvert cotxe.stl cotxe.obj
```

1

Els arxius ".obj" ocupen menys espai que els arxius ".STL".

12.3.1 Creació de la malla background

A continuació es crearà una malla amb blockMesh, dintre de l'arxiu blockMeshDict. Per tal de saber les dimensions es recomana utilitzar la funció sphere. Obrint el paraview

i obrint el cas corresponent. Com s'aplica a la figura 12.4, visualitzant els eixos (es troba a baix de tot, dintre de l'apartat de Properties) i utilitzant aquesta funció és més senzill poder generar la malla background, ja que es té una visualització molt clara d'on està ubicat cada punt inserint a mà (dintre de l'apartat de Properties) o arrossegant l'esfera amb el ratolí es poden localitzar els punts que l'usuari trobi oportú. Per accedir dintre d'aquesta funció cal anar a: Sources → Geometric Shapes → Sphere. En aquest cas es proposa els punts generats a l'arxiu blockMesh de l'**annex 1.4.3.1**. A part s'han de definir les diferents cares de l'hexaedre creat. Més tard s'utilitzaran per definir les condicions de contorn. En ser un únic hexaedre es reutilitzaran els noms del capítol 1 cas 1.

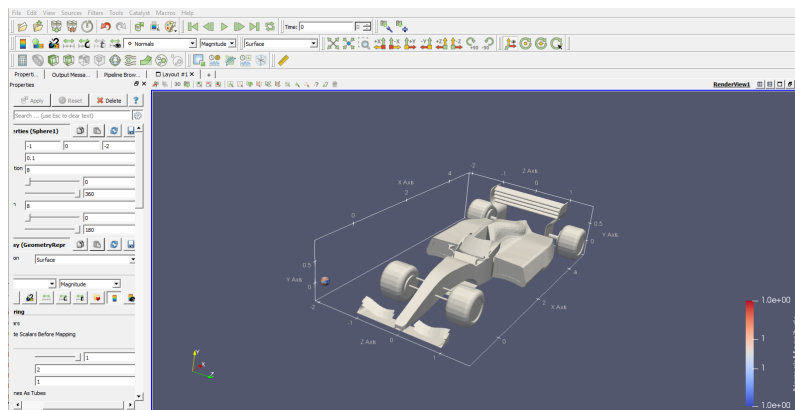


Figura 12.4: Utilitzant la funció sphere en el programa paraview

12.3.2 Creació del fitxer snappyHexMeshDict

Es pot copiar el mateix arxiu que el del capítol 3, sempre que es facin els canvis adequats. Canviant el nom de pipe.obj per al de cotxe.obj i eliminant totes les cares creades anteriorment (outlet, inlet i wall). Per defecte si només hi ha un objecte, es detecta com a la paraula solid.

A continuació també es modificaran els cells levels d'aquesta regió per tal de tenir una major precisió i poder crear cel·les més petites damunt de l'objecte se seleccionarà un nivell de (4 4) a mesura que augmenta aquest nombre, les cel·les de l'objecte disminueixen i vol dir que hi ha quatre menes de refinament de la malla.

Un altre apartat a canviar seria l'anomenat locationInMesh. Aquesta especificació on a continuació s'indicarà un punt. Indica on està localitzada la malla. Això vol dir si es desitja una malla interior on l'objecte és la malla cal indicar un punt on està dintre de la geometria creada (capítol 3). En canvi en aquest cas es desitja una malla exterior per tant simplement cal indicar un punt que se situï dintre de la malla i fora de l'objecte o solid. Per visualitzar-ho millor o si no es té clar on ubicar el punt es pot utilitzar de nou la funció sphere del Paraview. En aquest cas es proposa el punt (0 2 0). Com es pot observar a la següent imatge no toca el sòlid ni està fora de la malla. Es pot comprovar tot el codi sencer a l'**annex 1.4.3.5**.

12.3.3 Generació dels altres arxius

- System. Dintre de la carpeta System es poden observar diferents tipus de fitxers. De totes maneres es poden consultar a l'**annex 1.4.3**.
 - ▶ **blockMeshdict**, ja s'ha especificat anteriorment. A l'aparat 12.3 igual que l'arxiu snappyhexmesh 12.3.2. **Annex 1.4.3.1**.
 - ▶ **controlDict**, es pot utilitzar el mateix controlDict que el del capítol 3. **Annex 1.4.3.2**.
 - ▶ **fvSchemes** es pot utilitzar també el mateix. **Annex 1.4.3.3**.
 - ▶ **fvSolution**, en aquest apartat s'ha canviat un poc, ja que s'ha fet una malla una mica més fina tarda molt a produir les interaccions. Per tant s'ha decidit augmentat una mica el valor dels residuals mínims. Per exemple la pressió s'ha canviat a $2 \cdot 10^{-2}$. S'ha sacrificat la precisió de la pressió per obtenir menys temps de càlcul. Es pot modificar al gust de l'usuari, depenent de la precisió que l'usuari desitgi. També s'han canviat una mica els factors de relaxació tal com es va comentar anteriorment. **Annex 1.4.3.4**.
 - ▶ **meshQualityDict**, és el mateix arxiu que en el capítol 3.
 - ▶ **surfaceFeaturesDict**, simplement s'ha de canviar el nom de l'objecte. En aquest exemple té com a nom "cotxe.obj". **Annex 1.4.3.6**.
- Constant. Es pot consultar el codi a l'**annex 1.4.2**.
 - ▶ Dintre, hi ha dues carpetes dintre. TriSurface i polyMesh. Trisurface hi ha la geometria i dintre de polymesh si troba la informació de la malla, generat per blockMesh.
 - ▶ transportProperties, on es declararà com a model de transport newtonià i indicar la viscositat cinemàtica. **Annex 1.4.2.1**.
 - ▶ turbulenceProperties, s'utilitza la simulació de tipus RAS (Reynolds-Averaged Simulation). Amb un model de turbulència kOmegaSST. Indicant que és un model turbulent i que imprimesqui per pantalla els coeficients de turbulència. **Annex 1.4.2.2**.
- A la carpeta 0 s'hi han de determinar les variables de nut, p, U ,omega i k. Es pot consultar el codi a l'**annex 1.4.1**.
 - ▶ U, fent referència a la velocitat, es proposa una velocitat de 50 m/s, en direcció x. Que simula la velocitat del vent. Apareix una nova funció que fa referència la malla del sòlid que serà declarada com a noSlip, ja que se suposa que l'aire relisca damunt del sòlid. **Annex 1.4.1.5**.
 - ▶ p, la pressió a l'entrada (inlet) és desconeguda, per tant s'utilitza zeroGradient. La pressió a la sortida serà uniforme de 0. **Annex 1.4.1.4**
 - ▶ k, nut i omega són valors utilitzats per calcular el model turbulent, cosa que s'utilitzaran els mateixos que el tutorial d'Openfoam motorbike. Descarregant en el capítol 3. k es pot trobar a l'**annex 1.4.1.1**, nut a l'**annex 1.4.1.2** i omega a l'**annex 1.4.1.3**.

12.4. Visualització de la malla i execució del cas

Una vegada fets els canvis, corresponents es procedirà a crear la malla. Amb la funció snappyHexMesh. S'han de seguir els següents passos:

1. En el directori del cas 3. Utilitzar la funció blockMesh.
2. Utilitzar la funció snappyHexMesh.
3. Generat un arxiu ".foam" introduint en el directori, paraFoam -touch -builtin.
4. Obrir el paraview i obrir el cas de l'arxiu ".foam".
5. Dintre de la carpeta es podrà veure que hi ha dos "time steps". I es podran visualitzar diferents "malles" la carpeta 1 i la 2. Si es visualitzen des del paraview i només visualitzant el sòlid. Si es canvia el directori a visualitzar. Es poden observar els següents canvis. En aquest cas veu millor la diferència que no en el capítol 3. A la visualització en el paraView per tal de valorar el canvi d'un "timeStep" a un altre únicament es fa la visualització del sòlid, tal com es pot veure a les diferents imatges.

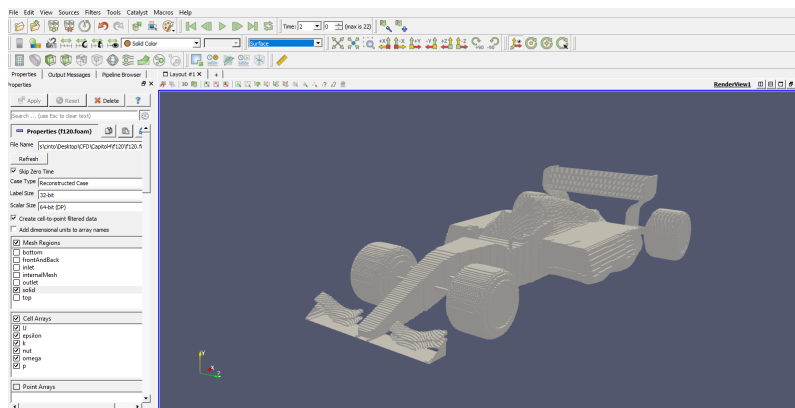


Figura 12.5: temps=1, generat per castellated mesh

En canvi un directori més endavant:

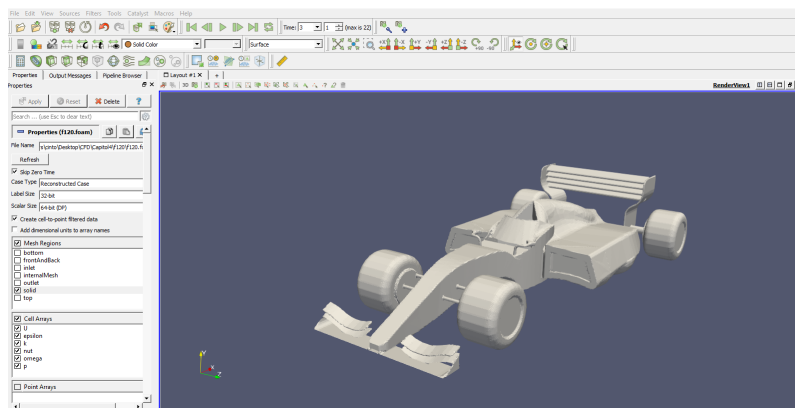


Figura 12.6: temps=2, generada per snappyHexMesh

6. A partir d'aquí ja es té la malla creada. Ara simplement cal canviar el nom del directori 0 per 0.org esborrar el directori 1, introduir dintre de la carpeta 2 tots els arxius de la nova carpeta 0.org i canviar el nom de la carpeta 2 per 0. Per tal que el programa pugui començar des de la carpeta 0, sinó generaria error.

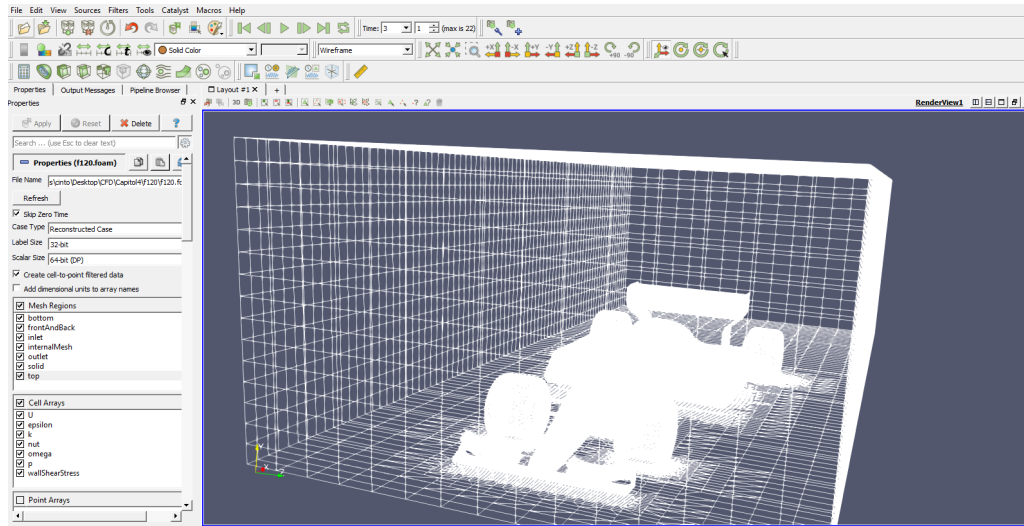


Figura 12.7: Malla completa

Cal recordar que si es vol esborrar i generar-ne una de nova cal esborrar tots els directoris i tornar a canviar 0.org per 0. A partir d'aquí es pot procedir a la resolució del cas utilitzant simpleFoam.

12.5. Postprocés

Una vegada ha acabat la simulació es poden determinar les següents dades. Es generarà un arxiu ".foam" per tal de visualitzar els resultats. A continuació apareixeran els resultats fent referència al camp de velocitats i al camp de pressions.

Apareixerà un hexaèdre que és la malla generada en blockMesh on a dintre hi haurà la geometria. A continuació. Per tal de visualitzar millor les zones de malla que interessin el que es farà, serà ocultar les regions que no interessin. El procediment és anar dintre del arxiu.foam a properties s'ha de seleccionar "bottom" i "solid" on seran les úniques zones a què es visualitzaran.

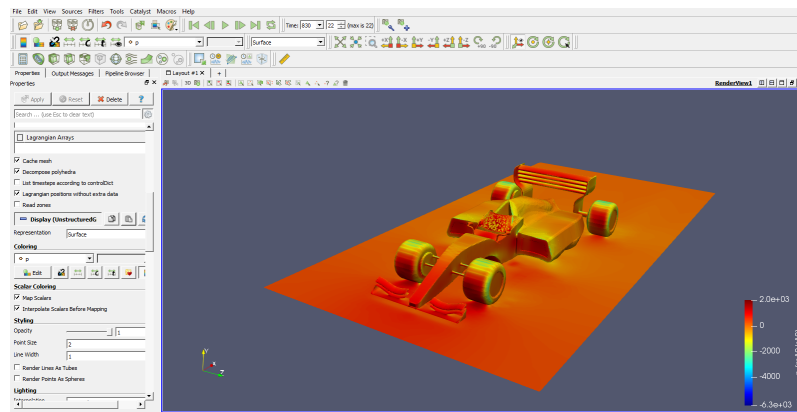


Figura 12.8: Camp de pressions damunt el sòlid F1

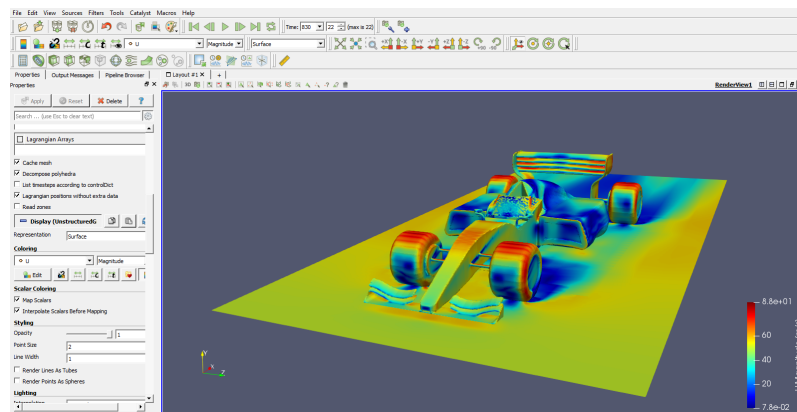


Figura 12.9: Camp de velocitats damunt el sòlid F1

12.5.1 Streamlines

Per tal de visualitzar les línies de corrent. Seguir els següents passos.

1. Cal tornar a visualitzar totes les regions de la malla. Tal com es veu a la figura que apareix a continuació.

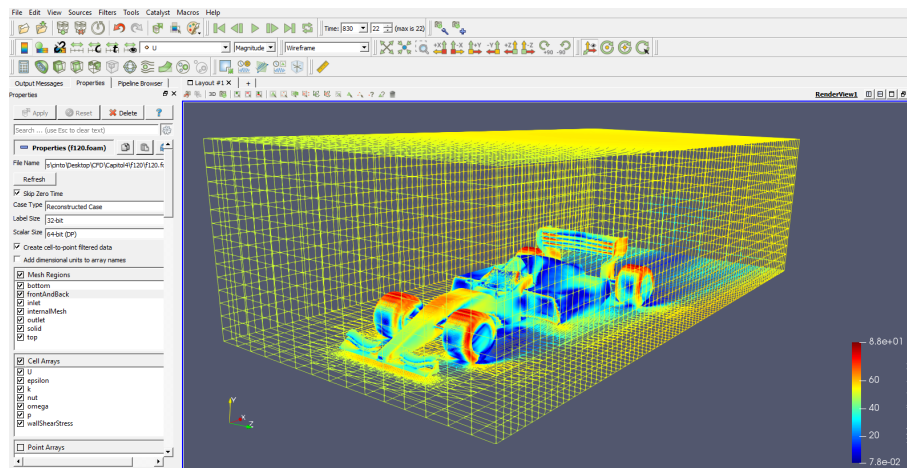


Figura 12.10: Visualització de la malla i la velocitat.

2. Una vegada s'ha creat, procedir a clicar el botó de Streamlines. Seguir el procediment indicat a la figura 12.11.

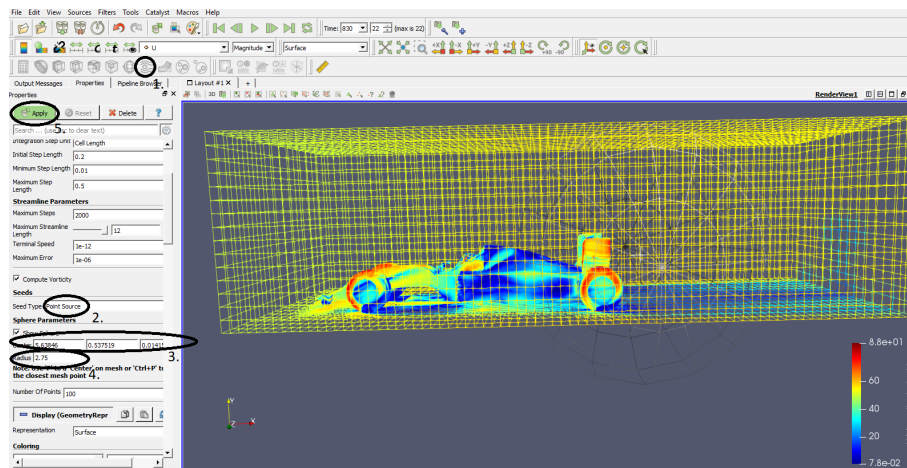


Figura 12.11: Procediment per crear línies de corrent amb Streamtracer

3. A continuació per obtenir una millor visualització de les línies de corrent. S'accedirà a Pipeline Browser i clicarà una vegada damunt l'arxiu ".foam" a continuació s'accedirà a filters → alphabetical → Extractblock.
4. Accedir a properties i a continuació seleccionar les regions bottom, solid i clicar apply.

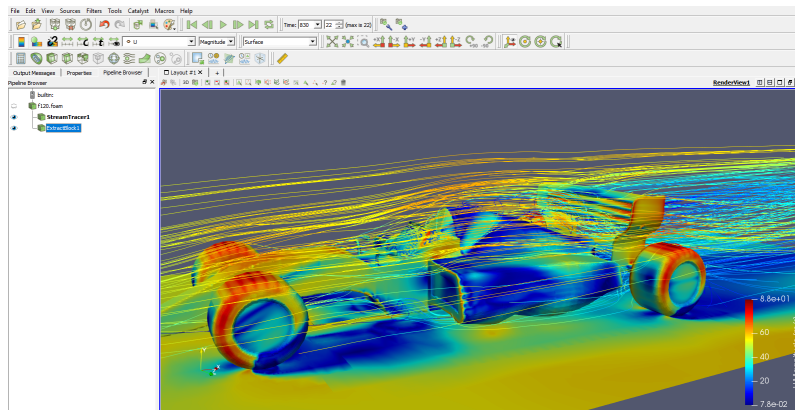


Figura 12.12: Visualització final de les línies de corrent

12.5.2 Càlcul de la força d'arrossegament i força de sustentació amb Paraview

És interessant fer càlculs amb paraview. Per tal de fer aquest càlcul s'agafarà cada cel·la on es generaran els vectors normals i a partir d'aquí multiplicant per la pressió en m^2/s^2 per tant s'ha de multiplicar la densitat de l'aire, que se suposa constant amb $1.204 \text{ kg}/m^3$. [28]

Per tant se seguirà, partint des de 0.

1. Primer obrir l'arxiu “.foam”. Una vegada obert utilitzar la funció extractBlock només s'ha de visualitzar la regió solid.

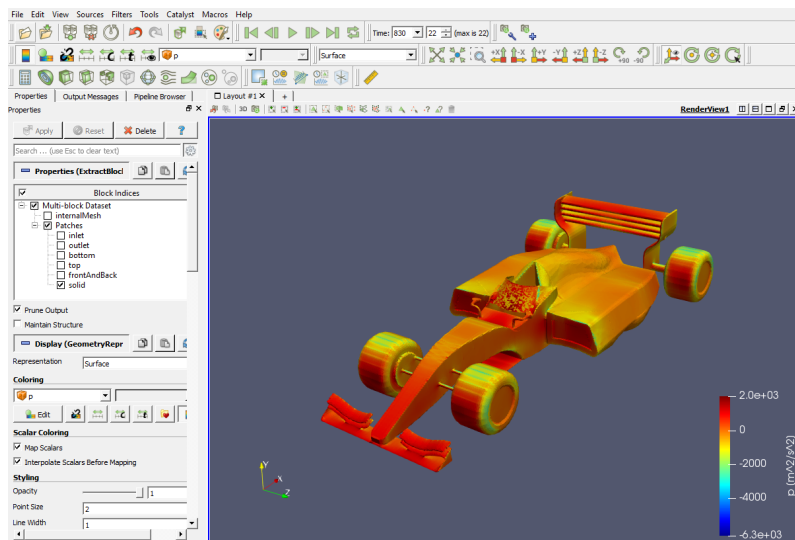


Figura 12.13: Ús de la funció extractBlock

2. A continuació utilitzar la funció ExtractSurface. S'hi pot accedir: filters → alpha-betical → ExtractSurface i dintre de “Properties clicar apply”.

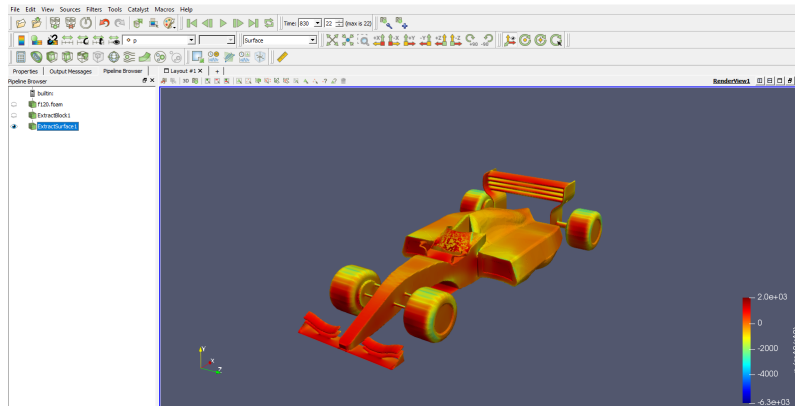


Figura 12.14: Funció ExtractSurface

3. Una vegada extreta la superfície, es genereràn els vectors normals de la superfície els quals es descomposen en els tres eixos. També s'ha seleccionat l'inversió de cel·les ja que els vectors de damunt l'aleró estan girats i donaria un lift positiu. S'ha d'utilitzar la funció “Generate surface normals”. A continuació s'ha de marcar l'opció “Compute Cell Normals” i es clica apply.

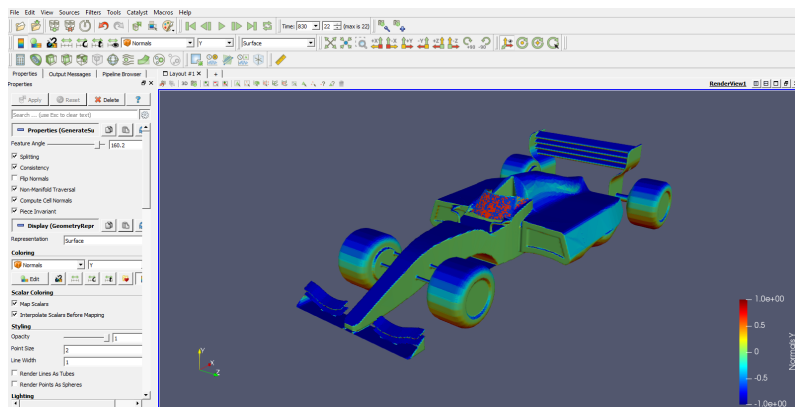


Figura 12.15: Generació del vector normal de les cel·les i visualització dels vectors en Y.

4. Una vegada realitzat el pas 3 cal utilitzar la funció “Calculator”. Per tal de calcular la força de sustentació o també anomenat lift. S'ha de multiplicar la pressió (escalar) per les normals amb Y i al ser la $\frac{p}{\rho}$ es multiplicarà per la densitat. A l'apartat de “Attribute Type” seleccionar Cell data.

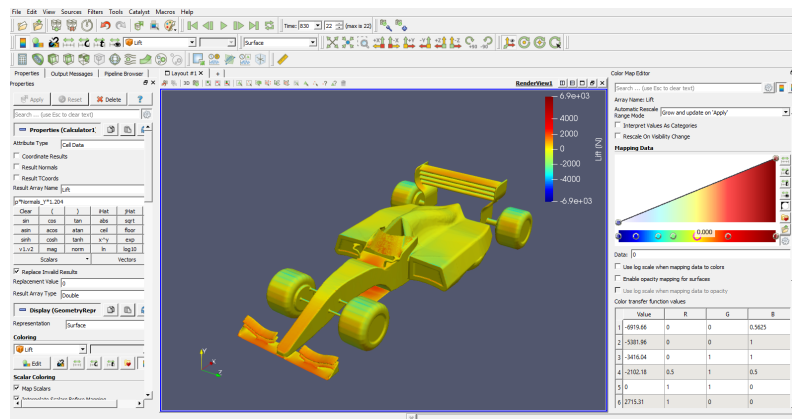


Figura 12.16: Visualització de la força de sustentació,

També a l'apartat de Coloring i clicant el botó d'edit apareixerà l'editor per canviar el mapa de colors. En aquest cas s'ha agafat la icona (seleccionat amb rosa) i s'ha establert com a referència del 0.

5. Ara es tenen tots els valors del lift. Simplement cal utilitzar la funció integrate magnitude per tal de tenir un únic valor. On es troba a filters → alphabetical → Integrate Variables.

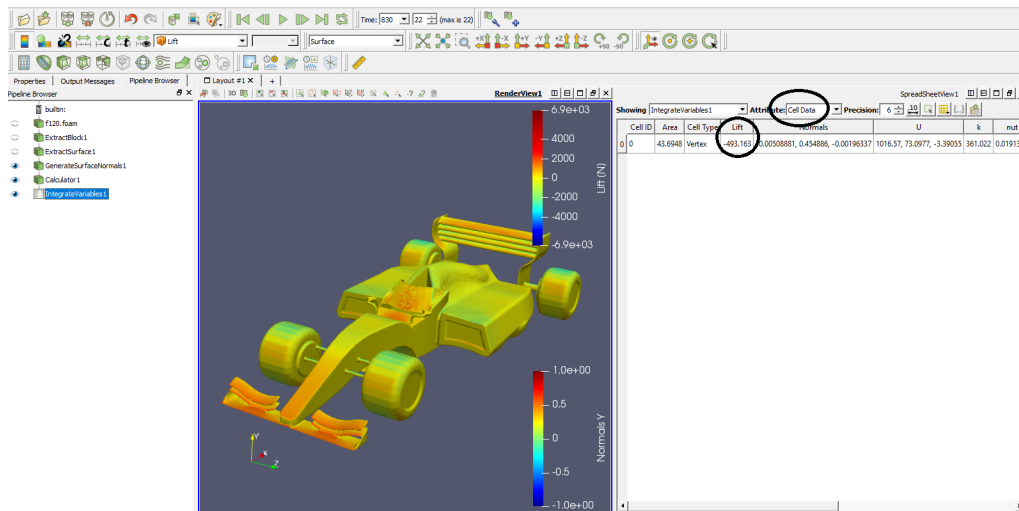


Figura 12.17: Força de sustentació en Newtons.

6. Es pot fer el mateix per a la força d'arrossegament. En aquest cas no s'han d'invertir, ja que si es visualitzen els vectors, tenen el sentit correcte. L'únic que varia és que s'utilitzaran els vectors normals en x en comptes dels vectors en y.

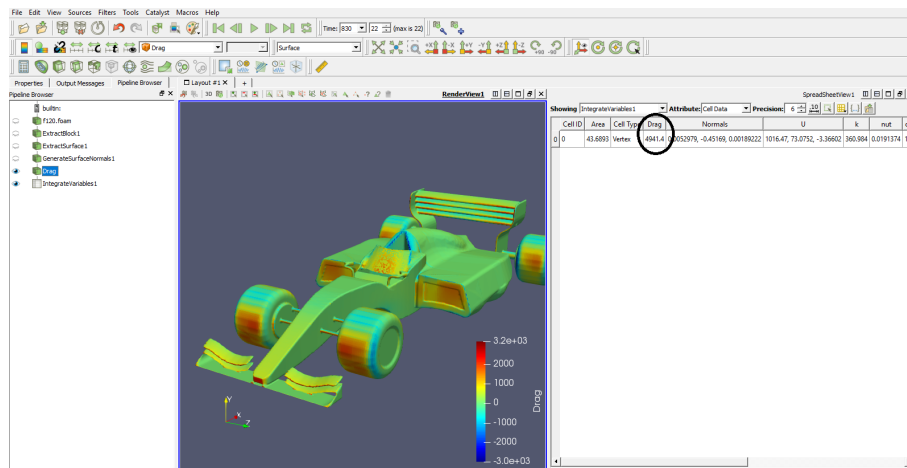


Figura 12.18: Força d'arrossegament en Newtons.

Les dues forces apareixen en newtons. Es pot extreure, la força d'arrossegament (drag) és de 4941 N i la força de sustentació és de -493 N. Com es pot veure encerclat en negre a les figures 12.17 i 12.18.

13. Síntesi de resultats

En aquest apartat una vegada acabats tots els capítols es farà una síntesi de cada cas.

- Capítol 1, flux entre plaques paral·leles.

- Cas 1, utilitzant la representació de vectors es comprova que els perfils de velocitats té una tendència lineal. Es pot observar que el resultat és correcte, ja que els valors límit són els imposats a les condicions de contorn.

- Cas 2, utilitzant la funció del paraview "plot over line" es pot observar que el perfil de velocitats descriu una paràbola i la pressió varia linealment tal com s'ha descrit a l'equació 9.23 i 9.21. També es comprova l'error entre el cabal teòric i el calculat per al programa i té un 0.28% i refinant la malla s'obté una precisió de 0.09% d'error. Utilitzant el solver simplefoam per al mateix cas per tal de calcular les tensions tangencials entre el fluid i la placa. Es comprova que és exactament el mateix resultat amb la figura 9.20 i l'equació 9.29.

- Cas 3, al tenir el moviment entre plaques en sentit contrari es pot observar que a la figura 9.25 i 9.26 són representades. La 9.25 com a magnitud i 9.26 com a component. A la comprovació del cabal hauria de ser $0 \text{ m}^3/\text{s}$, ja que es desplacen a la mateixa velocitat però en sentit contrari, però surt un cabal de $2.54 \cdot 10^{-20} \text{ m}^3/\text{s}$, que pràcticament podria ser 0.

- Capítol 2, flux per una canonada.

- Cas 1, utilitzant l'expressió de la velocitat màxima de forma analítica té com a resultat 0.975 m/s mentre que la resolució amb Openfoam és de 0.992 per tant l'error és d'un 1.74%. Fent referència al cabal que és una dada inicial es suposa $25 \text{ cm}^3/\text{s}$ i si es calcula amb el programa el resultat és d'uns $24.99 \text{ cm}^3/\text{s}$ amb un error de 0.0352%. Segurament fent un refinament de la malla l'error seria molt més baix.

- En el cas 2, comparant la velocitat màxima de l'expressió analítica amb la del programa. Els resultats són els següents 3.5760 m/s i 3.5705 m/s amb un error del 0.15%. En canvi el cabal amb un error de 0.013%.

- En el cas 3, els resultats no són tan exactes com es pot observar a les figures 10.25 i 10.26 com els altres casos però s'assimilen. On es té més error és als esforços tangencials que en aquest cas és d'un 17.16%. Es creu que aquest error és a causa del refinament de la malla, ja que en l'eix de les z únicament hi ha una cel·la i es calculen els valors extrems de la cel·la, un refinament de la malla amb una graduació igual que al capítol 2 en el cas 3. Es podria acurar i extreure millors resultats.

- Capítol 3, estudi d'un flux per un colze.

- En el primer cas és una comparativa entre resultats on entren un flux laminar i un flux turbulent. On es pot observar que les velocitats més altes s'aconsegueixen al flux laminar. A més, en el cas del flux turbulent perquè passi un flux a 5 m/s necessita un increment de pressió de 3465 Pa si a la pressió de sortida és 0 Pa .

- En el segon cas es comprova que sumant tots els cabals d'entrada i el de sortida el seu valor sigui 0, efectivament es compleix. És una comprovació senzilla però és interessant per a simulacions més complexes, tenir clar que de forma senzilla es poden fer comprovacions per saber si el programa ha fet una correcta simulació.
- Capítol 4, consisteix amb d'una simulació d'un flux que passa per un cotxe. S'han determinat, els camps de pressions i s'han fet el càlcul de la força d'arrossegament 4941N, representa la força que ha de fer el cotxe per mantenir-se a la velocitat a la qual s'està desplaçant. Una forma de baixar aquesta força seria obrir les entrades d'aire dels costats, ja que s'ha fet la simulació amb aquestes tancades. La força de sustentació -493 N, fa aproximadament 50kg de força cap a terra.

14. Implicació Mediambiental

En aquest apartat per tal de tenir en compte l'impacte mediambiental que ha tingut l'elaboració d'aquest estudi es farà la realització del càlcul de kg de CO_2 dispersats per l'atmosfera a causa d'aquest estudi o també conegut com a "carbon footprint". S'ha agafat de referència un document generat per l'Oficina Española de cambio climático en el 2016. [20]

Hi ha dos tipus de paràmetres que comptabilitzen les emissions, d'abast 1 (coneguts també com a directes) que aquests fan referència a desplaçament de vehicles, refrigeració, etc.) que en aquest estudi no s'han utilitzat. El d'abast 2 (coneguts com a indirectes), fan referència al consum elèctric.

Per tant en aquest estudi ha fet falta un portàtil que té un carregador de 120W que ha estat sempre connectat a la xarxa elèctrica. Si del projecte s'han dedicat 635 hores, el factor es pot consultar a la taula de l'**annex 1.5** [21] i la companyia elèctrica és Endesa.

Taula 14.1: Càlcul dels kg CO_2

Activitat	Dedicació [h]	[kW]	[kWh]	Factor d'impacte [<i>kg de CO_2</i> /kWh]	[<i>kg de CO_2</i>]
Portàtil	615	0.120	73.8	0.38	28.04

15. Planificació

Per tal de complir amb el calendari és de vital importància una bona planificació. Per això a continuació a la imatge 15.1 s'adjunta el nombre de tasques que s'han seguit durant el període de desenvolupament d'aquest estudi.

Tarea

Nombre	Fecha de inicio	Fecha de fin
Recerca d'informació i recerca de casos	4/02/19	22/02/19
Capítol 1.1: Plaques paral·leles amb moviment relatiu, funcions bàsiques de pre-processing i de post-processing	25/02/19	1/03/19
Capítol 1.2 : Crear diferents casos de les dues plaques paral·leles, moviment d'una placa, gradient amb plaques fixes i moviment de les dues plaques.	4/03/19	4/03/19
Redacció de funcions, pre-process i post-proces bàsic del capítol 1.1 i del capítol 1.2 del preprocès	5/03/19	8/03/19
Realització del project charter	4/03/19	7/03/19
Redacció del Capítol 1.2: Explicació de funcions com el càlcul del cabal , extracció del procés de perfils i forces tangencials provocades per el fluid i Plantejament del cas 3	11/03/19	14/03/19
Capítol 1 cas3: Execució del problema, i execució de funcions de postprocès	15/03/19	19/03/19
Execució del capítol 2: Estudi del flux que passa per una canonada 2d, realització de malla.	20/03/19	22/03/19
Recerca d'informació i instal·lació d'Openfoam i del programa gedit	25/03/19	29/03/19
Recerca d'informació i instal·lació de programari complementari com Paraview i Xming	1/04/19	4/04/19
Acabar el cas 1 del capítol 2: Explicant funcions com la wedge, inletOutlet, generació d'arcs per blockMesh, localització de valors amb paraview i residuals	5/04/19	10/04/19
Execució del cas 2 capítol 2, generació d'una malla per una canonada anular, realització del problema de forma anàlítica.	11/04/19	17/04/19
Comprovació de resultats analítics del Cas 2 i comprovació de caudal i velocitat. Introducció dintre del cas 3	18/04/19	23/04/19
Execució del cas 2.3, estudi de una canonada anular amb un cilindre dintre que gira.	24/04/19	1/05/19
Execució del problema i redacció del cas 3 capítol 2: explicació de la funció rotatingwall, representació amb gnuplot i funció "components".	2/05/19	8/05/19
Capítol 3 execució i redacció : comparativa amb un model turbulent i model laminar d'un colze 3d generat amb la funció snappyHexMesh i utilització del programa Salome.	9/05/19	13/05/19
Ensamblatge del document i redacció del document	14/05/19	20/05/19
Comprovació de documentació i petits retocs	21/05/19	3/06/19

Figura 15.1: Llistat de tasques

Una vegada s'han exposat el nombre de tasques es generarà el diagrama de Gantt. Que s'exposa a continuació.15.2

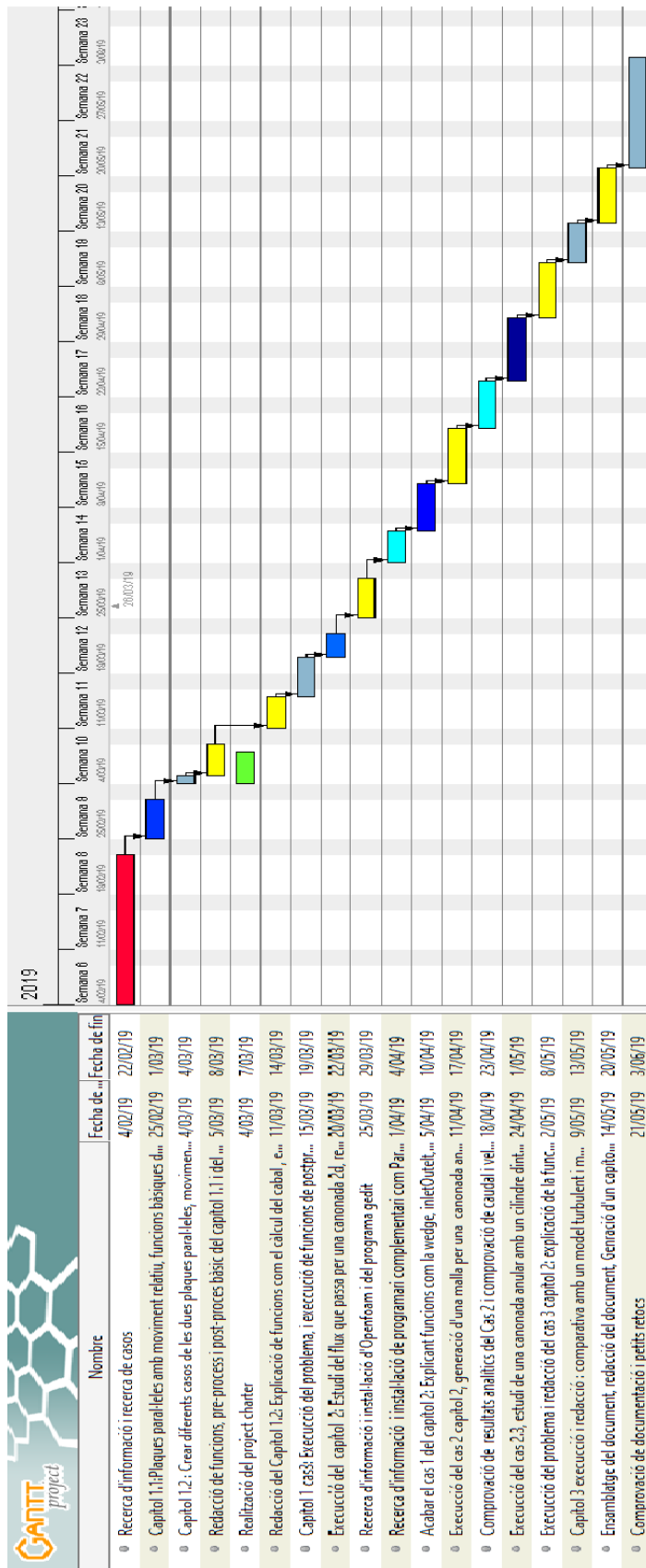


Figura 15.2: Diagrama de Gantt

Per tal d'entendre la gestió del temps dedicat s'adjunten:

- La densitat de treball amb cada setmana les hores dedicades:

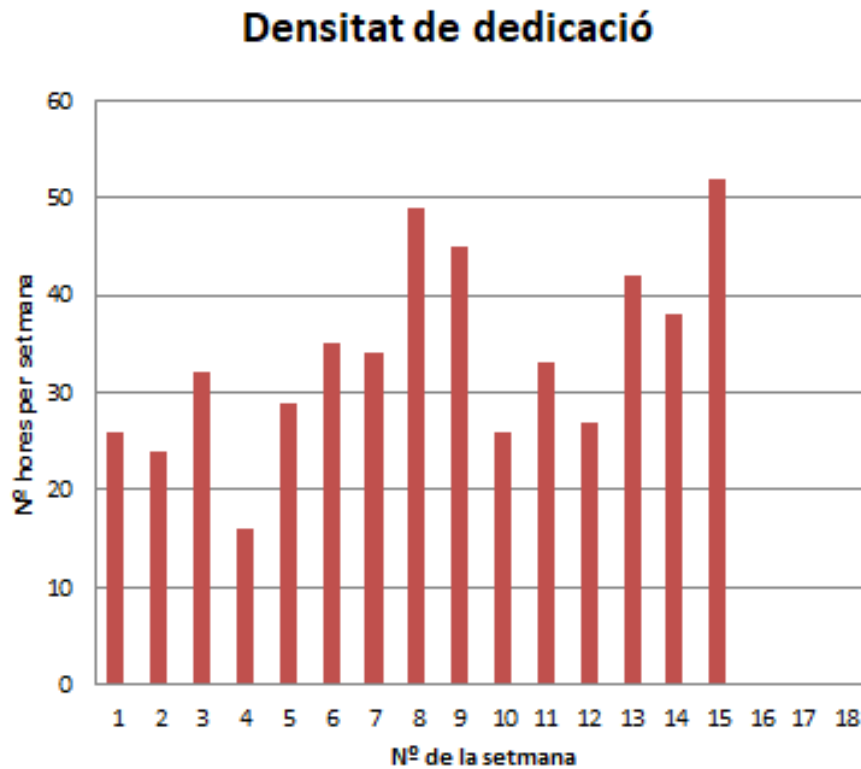


Figura 15.3: Gràfica de la densitat de treball

- Dedicació per grups:

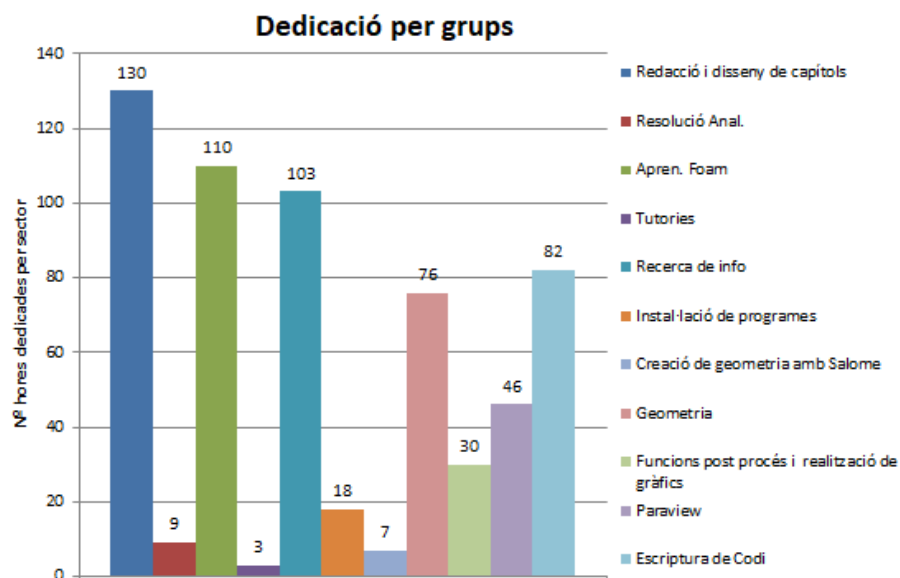


Figura 15.4: Dedicació d'hores per grups

16. Pressupost

En aquest apartat es durà a terme la quantificació econòmica de l'estudi realitzat. S'ha desglossat amb diferents apartats, per tal d'obtenir una millor comprensió.

A continuació, la taula 16.1 fa referència a la quantificació del temps i econòmica dedicada dintre de l'àmbit de la gestió d'informació.

Taula 16.1: Referent a la quantificació econòmica a la gestió de la informació

Gestió d'informació	Temps dedicat [h]	Preu per hora[€/h]	Preu total [€]
Recerca d'informació	103	15	1545
Tutories	3	0	0
Execució de les solucions analítiques	9	15	135
Aprenentatge OpenFoam	110	15	1650
Redacció i disseny de capítols	130	10	1300
Instal·lació del programari	18	15	270
TOTAL	373	-	4900

La taula 16.2 té com a funció l'elaboració de la quantificació de la dedicació i de l'import econòmic del pre-procés de forma desglossada.

Taula 16.2: Referent a la quantificació econòmica dels pre-processos

Preprocessos	Temps dedicat [h]	Preu per hora[€/h]	Preu total [€]
Geometria i generació de les malles	76	15	1140
Escriptura del codi	82	15	1230
Generació de geometria amb Salome	7	15	105
TOTAL	165	-	2475

La taula 16.3 fa referència als conceptes i al temps dedicats a l'elaboració dels post-processos.

Taula 16.3: Referent als postprocessos

Postprocessos	Temps dedicat [h]	Preu per hora[€/h]	Preu total [€]
Funcions postprocès i realització de gràfics	30	15.00	450
Utilització de resultats amb Paraview	46	15.00	690
TOTAL	76	-	1140

La taula resum amb suma de tots els àmbits de dedicació 16.4.

Taula 16.4: Total del pressupost

Grups d'activitats	Temps dedicat [h]	Preu total [€]
Gestió de la informació	373	4900
Preprocessos	165	2475
Postprocessos	76	1140
Subtotal del projecte	614	8515
	IVA 21 %	1788.15
TOTAL	614	10304

17. Futurs projectes

Una vegada realitzada la guia bàsica per utilitzar els programes de codi lliure (open-source) per l'anàlisi de la dinàmica de fluids, com bé s'ha dit apareixen les funcions simples i més utilitzades. Però per res queda acabada aquesta feina, hi ha infinitat de coses a explicar sobre aquests programes, per tant es proposa seguir amb la guia utilitzant funcions més complexes com serien:

- Elaboració de malles, utilitzant altres programes, diferents formes de crear malles o malles dinàmiques per exemple la simulació del funcionament d'un amortidor, pistó, etc.
- Ús d'altres "solvers", com serien solvers multifase, solvers que admeten fluxos compressibles, fluxos turbulents transitoris, etc.
- O fins i tot altres mòduls de dintre de l'OpenFoam, com seria el mòdul d'stress, combustió, transferència de calor, etc.

I moltes coses més, com es pot veure dintre de l'àmbit de la mecànica de fluids computacional inclou un gran ventall d'opcions i d'adaptar-se a casos molt concrets, que amb resolucions analítiques ara mateix és impensable.

18. Conclusions

L'execució d'aquest projecte, fa una introducció dintre de les bases del CFD. Per introduir a alumnes nous dintre d'aquest àmbit, perquè no es faci tan costosa l'arribada d'aquest nou món, per tal que amb les bases procedents d'aquest estudi es consolidi el fonament per poder seguir el seu propi camí dintre d'aquest àmbit i l'usuari obtingui major autonomia en el maneig d'aquest programa.

Per tal de formalitzar aquesta guia, s'ha generat un capítol per la instal·lació de programari i 4 capítols amb casos explicats. El recorregut d'aquest estudi ha estat el següent:

1. Per començar s'ha explicat la instal·lació del programa amb Windows i tots els programes complementaris. Donant així més accessibilitat a les persones que vulguin començar, ja que gran quantitat d'usuaris tenen Windows. Openfoam és un programa creat per ser utilitzat en Linux.
2. El primer capítol és una introducció perquè l'usuari pugui començar a interactuar amb el format que ofereix Openfoam. S'explica detall per detall la interpretació de cada un dels fitxers i l'ús de les funcions corresponents. Explicant com es crea una malla amb blockMesh, explicant com refinar-la, interpretar diferents funcions per les condicions inicials, propietats del fluid i així adaptar el cas al problema proposat. En la part del postprocés s'explica la visualització de resultats, la representació de diferents magnituds, vectors i finalment les funcions de postproces com serien les funcions del càlcul del cabal, explicació de com utilitzar altres solvers per calcular esforços tangencials. S'ha adaptat el codi per la versió v6.
3. El segon capítol, s'ha explicat com generar arcs i punts coincidents per la malla amb la funció blockMesh. L'ús de funcions com inletOutlet, o també la funció rotating. La localització de valors una vegada simulat amb paraview. La representació de resultats amb gnuplot per tal de comparar amb la resolució analítica i també la representació dels residuals per tal d'estudiar la convergència de la solució.
4. El tercer capítol, s'explica com generar una geometria amb el programa Salome utilitzant diferents funcions com el dibuix 2d, l'extrucció i el fusionat d'objectes. També el mallat amb la funció SnappyHexMesh. La utilitat de la funció sphere per agafar mesures del paraview per tal de crear una malla "background". L'ús de les funcions amb paraview de clip, slice i com interpretar dos resultats a l'hora de diferents exemples. Referent a les funcions de postproces s'utilitza la funció de "patchaverage". Finalment també s'explica un exemple de no convergència i un mètode per solucionar-lo.
5. En el quart capítol és generada una malla externa on es fa la simulació d'un cotxe i s'explica el procediment a seguir. Llavors són visualitzades les línies de corrent, el càlcul de forces com la força de sustentació i força d'arrossegament.

En tots aquests capítols com s'ha justificat anteriorment, es creu haver complit amb l'objectiu i també l'abast del projecte.

S'observa que els programes “opensource” són de gran qualitat, no estan en desús sinó just el contrari que de cada vegada agafen més força i es depenen d'ells. Concretament els utilitzats en aquest estudi no són senzills d'utilitzar però tenen un gran avantatge, és que són molt flexibles, això proporciona una adaptabilitat als problemes que es proposen. Un altre avantatge és que el programa amb la llicència que té, permet ser distribuït, modificat i estudiar el seu codi font. Això proporciona, un altre avantatge que és el seu ràpid desenvolupament.

Al meu parèixer, dintre de l'àmbit de l'enginyeria mecànica (i altres més) és una bona eina, ja que a part dona una millor visió pràctica a la mecànica de fluids. Per altra banda, serveix per calcular problemes amb gran dificultat que de forma analítica seria impossible. A part, en els estudis de grau en aquest cas als d'enginyeria mecànica, no es fa referència en aquests coneixements cosa que aquest estudi intenta suplir. Demostrant així que l'ús de programari lliure per resoldre problemes d'enginyeria és una opció que s'ha de contemplar sempre.

Bibliografia

- [1] Jordi Casacuberta Puig, J. (2014). The Foam House. [online] Thefoamhouse.es. Available at: <https://www.thefoamhouse.es/> [Accessed 15 Mar. 2019].
- [2] White, F. (2003). Fluid mechanics. New York: McGraw-Hill.
- [3] Bergadà Granyó, J. (2015). Breve introducción teórica con problemas resueltos. 2nd ed. Barcelona: Oficina de Publicacions Acadèmiques Digitals de la UPC, pp.355-389.
- [4] C.L. Greenshields "OpenFOAM User Guide Version 6", OpenFOAM Found.Ltd [Online]. July 2017. [Consulta: 25 Febrero 2019]. Disponible en: <https://cfd.direct/openfoam/user-guide/>
- [5] I.Pérez García, *Estudio Aerodinámico del Flujo alrededor de un Ahmed Body mediante OpenFOAM*, "UPC,2016.
- [6] Jordi Casacuberta Puig, J. (2014). The Foam House. [online] Thefoamhouse.es. Available at: <https://www.thefoamhouse.es/> [Accessed 20 Feb. 2019].
- [7] Cfd-online.com. (2019). CFD Online. [online] Available at: <https://www.cfd-online.com/> [Accessed 4 Apr. 2019].
- [8] CFD Direct. (2019). OpenFOAM v6 User Guide: 5.4 Meshing with snappyHexMesh. [online] Available at: <https://cfd.direct/openfoam/user-guide/v6-snappyhexmesh/> [Accessed 16 May 2019].
- [9] Cfdsupport *TFCD reg. is focused and unlimited CFD software* fotografia [Online].[Consulta 5 de març 2019]. Disponible a: <https://www.cfdsupport.com/tcfd.html>.
- [10] Woldynamics.com. (2019). [online] Available at: <http://www.woldynamics.com/wiki/OFTipsandTricks.pdf> [Accessed 2 May. 2019].
- [11] Diversitat de programari [Online]. Consulta [06/03/2019]. Disponible a: <https://www.gestor-energetico.com/simulacion-energetica-cuales-los-mejores-software-simulacion-cfd/>
- [12] C.L. Greenshields "About OpenFOAM". [Online]. Consulta[06/03/2019]. Disponible a: <https://cfd.direct/openfoam/about>
- [13] D.Numerical and E.Simualtion *lli. introduccion computacional*, "pp.1-6. [Online]. México: 17/11/2017. [Consulta: 21 Febrero 2019]. Disponible en: http://bibing.us.es/proyectos/abreproy/4119/fichero/Proyecto_PDF+++Estudio+del+Sistema+de+Ventilaci
- [14] Greenshields, C. (2019). Download v6 — Ubuntu. [online] OpenFOAM. Available at: <https://openfoam.org/download/6-ubuntu/> [Accessed 26 Mar. 2019].

- [15] Paraview.org. (2019). Download — ParaView. [online] Available at: <https://www.paraview.org/download/> [Accessed 26 Mar. 2019].
- [16] Harrison, C. (2019). Xming X Server for Windows - Official Website. [online] Straightrunning.com. Available at: <http://www.straightrunning.com/XmingNotes/> [Accessed 16 May 2019].
- [17] Salome-platform.org. (2019). Download Page — SALOME Platform. [online] Available at: <https://www.salome-platform.org/downloads/current-version> [Accessed 16 May 2019].
- [18] Microsoft Store. (2019). Get Ubuntu - Microsoft Store. [online] Available at: <https://www.microsoft.com/en-us/p/ubuntu/9nblggh4msv6?activetab=pivot:overviewtab> [Accessed 9 Mar. 2019].
- [19] SL, U. (2019). gedit (Windows). [online] Uptodown.com. Available at: <https://gedit.uptodown.com/windows> [Accessed 6 Mar. 2019].
- [20] Guía para el cálculo de la huella de carbono y para la elaboración de un plan de mejora de una organización. (2017). 3rd ed. [PDF] Madrid: Oficina Española de Cambio Climático, Ministerio de agricultura, alimentación y medio ambiente, pp.2-40. Available at: https://www.miteco.gob.es/es/cambio-climatico/temas/mitigacion-politicas-y-medidas/guia_huella_carbono_tcm30-178893.pdf [Accessed 17 May 2019].
- [21] MixComerc y factores impacto. (2018). [PDF] CNMC (Comisión Nacional de los Mercados y la Competencia, p.10. Available at: <https://gdo.cnmc.es/CNE/resumenGdo.do?anio=2018> [Accessed 17 May 2019].
- [22] Hanna, K. (2019). Market Trends « Keith Hanna. [online] Blogs.mentor.com. Available at: <https://blogs.mentor.com/khanna/blog/tag/market-trends/> [Accessed 21 May 2019].
- [23] Resolved Analytics CFD Consulting and Simulation Strategy. (2019). Comparing CFD Software. [online] Available at: <https://www.resolvedanalytics.com/theflux/comparing-cfd-software> [Accessed 21 May 2019].
- [24] F. Moukalled, L. Mangani and M. Darwish (2016), “The Finite Volume Method in Computational Fluid Dynamics. An advanced introduction with OpenFOAM and Matlab”, Springer
- [25] grabcad.(2019). ASTA F1 CAR. [online] Available: <https://grabcad.com/library/asta-f1-car-1>
- [26] Nagy, J. (2019). How to install OpenFOAM and run a simulation in Windows 10 - tutorial. [online] YouTube. Available at: https://www.youtube.com/watch?v=xj0dB_PsElg&t=647s [Accessed 5 Apr. 2019].
- [27] Sanchez Sopena, Á. (2017). Study of the interaction of the rear wing and diffuser flows in a Formula Student car. 1st ed. Terrassa, p.18.
- [28] YouTube. (2019). How to calculate aerodynamic forces with Paraview. [online] Available at: https://www.youtube.com/watch?v=J944H0j_4b0 [Accessed 20 May 2019].

- [29] Albertopassalacqua.com. (2019). Developing for OpenFOAM on Windows 10 – Part 1 – Alberto Passalacqua. [online] Available at: <https://www.albertopassalacqua.com/?p=1627> [Accessed 10 Apr. 2019].